ON INVARIANT FORMULAE OF FIRST-ORDER LOGIC WITH NUMERICAL PREDICATES

DISSERTATION
zur Erlangung des akademischen Grades

doctor rerum naturalium
(Dr. rer. nat.)
im Fach Informatik

eingereicht an der
Mathematisch-Naturwissenschaftlichen Fakultät
der Humboldt-Universität zu Berlin

von
M.Sc. Frederik Harwath

Präsidentin der Humboldt-Universität zu Berlin:
Prof. Dr.-Ing. Dr. Sabine Kunst

Dekan der Mathematisch-Naturwissenschaftlichen Fakultät:
Prof. Dr. Elmar Kulke

Gutachter/innen:
1. Prof. Dr. Nicole Schweikardt, Humboldt Universität zu Berlin
2. Prof. Dr. Stephan Kreutzer, Technische Universität Berlin
3. Dr. Luc Segoufin, INRIA, LSV, ENS de Cachan

Tag der mündlichen Prüfung: 04.04.2018

Abstract

This thesis studies the concept of *order-invariance* of formulae of first-order logic (FO) and some of its extensions as well as other closely related concepts from finite model theory. Many results in finite model theory assume that structures are equipped with an embedding of their universe into an initial segment of the natural numbers. This allows to transfer arbitrary relations (e.g. linear order) and operations (e.g. addition, multiplication) on the natural numbers to the structure. The arising relations on the structure are called *numerical predicates*. If a formula uses these numerical predicates, it is often desirable to ensure that it defines a property of the underlying structure instead of a property of the embedding of the universe of the structure into the natural numbers. To this end, one considers formulae whose truth value in finite structures is *invariant* under changes to the embeddings of the structures. For instance, if the allowed numerical predicates include only a linear order, such formulae are called *order-invariant*. We study the effect which the invariant use of different kinds of numerical predicates has on the expressive power of FO and different extensions thereof. The results of this thesis can be divided into three parts.

In the first part, we consider formulae of first-order logic with modulo-counting quantifiers (FO+MOD) which may use arbitrary numerical predicates in an invariant way ($\mathcal{ARB}$-invariant formulae). A well-known restriction of FO is its *locality*. We study the locality and non-locality properties of $\mathcal{ARB}$-invariant FO+MOD-formulae.

In the second part, we consider sentences of first-order logic (FO) which may use a linear order and the corresponding addition in an invariant way (addition-invariant formulae). We investigate the expressive power of such formulae on finite trees. We obtain a characterisation of the regular tree languages which can be defined by addition-invariant FO-sentences on finite trees: these are exactly the tree languages which are definable by plain FO-sentences (i.e. sentences without numerical predicates) with certain cardinality predicates. To this end, we obtain a characterisation of the tree languages definable in this logic in terms of algebraic operations on trees.

In the third part, we compare the expressive power and the succinctness of different extensions of FO on structures of *bounded tree-depth*. In particular, we consider monadic second-order logic (MSO). It is known that FO and MSO have the same expressive power on structures of bounded tree-depth. We study the succinctness of MSO and FO on such structures and obtain essentially optimal upper bounds for the size of equivalent FO-sentences for given MSO-sentences. We show that order-invariant MSO has the same expressive power as plain FO+MOD and that order-invariant FO has the same expressive power as plain FO on structures of bounded tree-depth and we compare the succinctness of these logics.

Zusammenfassung

Diese Arbeit untersucht das Konzept *ordnungsinvarianter* Formeln der Logik erster Stufe (FO) und einiger ihrer Erweiterungen, sowie andere eng verwandte Konzepte der endlichen Modelltheorie. Viele Resultate der endlichen Modelltheorie nehmen an, dass Strukturen mit einer Einbettung ihres Universums in ein Anfangsstück der natürlichen Zahlen ausgestattet sind. Dies erlaubt es, beliebige Relationen (z.B. eine lineare Ordnung) und Operationen (z.B. Addition, Multiplikation) von den natürlichen Zahlen auf solche Strukturen zu übertragen. Die resultierenden Relationen auf den endlichen Strukturen werden als *numerische Prädikate* bezeichnet. Wenn eine Formel numerische Prädikate nutzt, ist es oftmals erstrebenswert, sicher zu stellen, dass die Formel lediglich eine Eigenschaft der zugrundeliegenden Struktur beschreibt, statt eine Eigenschaft der Einbettung der Struktur in die natürlichen Zahlen. Zu diesem Zweck betrachtet man Formeln, deren Wahrheitswert auf endlichen Strukturen *invariant* unter Änderungen der Einbettung dieser Strukturen ist. Wenn das einzige verwendete numerische Prädikat eine lineare Ordnung ist, spricht man beispielsweise von *ordnungsinvarianten Formeln*. Wir untersuchen die Auswirkung, die der invariante Gebrauch numerischer Prädikate auf die Ausdrucksstärke von FO und unterschiedlichen Erweiterungen von FO hat. Die Resultate dieser Arbeit können in drei Teile unterteilt werden.

Der erste Teil beschäftigt sich mit der Erweiterung von FO um *Modulo-Zählquantoren* (FO+MOD) und Formeln, die beliebige numerische Prädikate auf invariante Weise nutzen dürfen ($\mathcal{ARB}$-invariante Formeln). Eine bekannte Einschränkung von FO-Formeln ist ihre *Lokalität*. Wir untersuchen die Lokalitätseigenschaften $\mathcal{ARB}$-invarianter FO+MOD-Formeln.

Im zweiten Teil beschäftigen wir uns mit FO-Formeln, die eine lineare Ordnung samt der zugehörigen Addition auf invariante Weise nutzen dürfen (additionsinvariante Formeln). Wir untersuchen die Ausdrucksstärke solcher Formeln auf endlichen Bäumen. Dabei erhalten wir eine Charakterisierung der regulären Baumsprachen, die von additionsinvarianten FO-Sätzen auf endlichen Bäumen definiert werden können: Dies sind genau die Baumsprachen, die durch einfache FO-Sätze (d.h. FO-Sätze, die keine numerischen Prädikate nutzen) mit bestimmten Kardinalitätsprädikaten definiert werden. Zu diesem Zweck entwickeln wir eine algebraische Charakterisierung der in dieser Logik definierbaren Baumsprachen durch Operationen auf Bäumen.

Der dritte Teil der Arbeit beschäftigt sich mit der Ausdrucksstärke und der Prägnanz von FO und Erweiterungen von FO auf Klassen von Strukturen *beschränkter Baumtiefe*. Insbesondere betrachten wir hierbei die monadische Logik zweiter Stufe (MSO). Es ist bekannt, dass FO und MSO auf Klassen von Strukturen beschränkter Baumtiefe die gleiche Ausdrucksstärke haben. Wir vergleichen die *Prägnanz* von MSO und FO, d.h. die Frage, wie groß die äquivalenten FO-Formeln für gegebene MSO-Formeln sind, und erhalten hierbei im Wesentlichen optimale obere Schranken. Wir zeigen, dass ordnungsinvariante MSO- bzw. FO-Formeln auf Strukturklassen beschränkter Baumtiefe die gleiche Ausdrucksstärke wie einfache FO+MOD- bzw. FO-Formeln haben und wir vergleichen die Prägnanz dieser Logiken.

Preface

I want to thank my advisor Nicole Schweikardt for introducing me to the subjects of this thesis as well as for her support, advice, and patience. I also want to thank her and my colleagues for the friendly work environment which I enjoyed for so long.

The results of this thesis are based on the results of the papers [HS12], [HS13], [HS16] which have been obtained in collaboration with Nicole Schweikardt and on the results of the paper [EEH14] which have been obtained with Michael Elberfeld and Kord Eickmeyer. During the time when I worked on this thesis, I have also collaborated with Lucas Heimberg and Nicole Schweikardt [HHS14], [HHS15], with Thomas Zeume [ZH16], and with Isolde Adler on results which have not found their way into this thesis. I want to thank Isolde, Kord, Lucas, Michael, Nicole, and Thomas for these collaborations. I am also grateful for all the helpful discussions with other researchers and for the work of the anonymous referees of my papers which often helped to improve these results.

I also want to thank my wife Nicole Harwath for being who she is and for tolerating the absentmindedness that my work sometimes causes. Specials thanks go to my children Amalia, Clemens, and Valentin for reminding me that there are other things in life than logic.

Contents

Introduction

Mathematical logic plays an important role in different subfields of computer science such as artificial intelligence, programming language semantics, formal verification, database theory, and complexity theory (see, for instance, [HHI$^+$07]), to name a few. The main topic of this thesis are formulae of *first-order logic* (FO) and some of its extensions on finite relational structures which are equipped with a total order. We study so called *order-invariant* formulae and several extensions of the concept of order-invariance. This topic is most closely connected with the last two of the mentioned subfields of computer science: database theory and computational complexity theory.

Database theory. The theory of *relational databases* has been linked with mathematical logic right from its inception when Codd [Cod70] introduced the relational model and introduced the relational calculus [Cod72] as a language to formulate queries to a database system. The relational calculus is essentially equivalent to FO and to understand its capabilities as a query language amounts to understanding the expressive power of FO on finite structures. One of the central goals of the relational model was, to quote from [Cod70],

> *data independence* - the independence of application programs and terminal activities from growth in data types and changes in data representation (...)

Codd names *ordering dependence* as the first in his list of sources of data dependence in database systems. To quote again from [Cod70],

> *Ordering dependence* - (...) Let us consider those existing systems which either require or permit data elements to be stored in at least one total ordering which is closely associated with the hardware-determined ordering of addresses. (...) Such systems normally permit application programs to assume that the order of presentation of records from such a file is identical to (or is a subordering of) the stored ordering. Those application programs which take advantage of the stored ordering of a file are likely to fail to operate correctly if for some reason it becomes necessary to replace that ordering by a different one.

To achieve the goal of data independence, at first sight, it might seem necessary to hide the total ordering which arises from the storage of the database from the query language. It is, by now, well-known that the expressive power of FO on finite structures is rather limited.[1]

[1]See [Imm99], [EF99], [Lib04] for modern introductions to the subject.

For instance, FO *cannot count*, e.g. FO cannot express that the number of elements of a set is even [Ajt83]. FO also cannot express the transitive closure of a binary relation [AU79] and, more generally, FO can only express *local properties* [Han65, Gai82]. This limitation of FO also applies to extensions with different forms of counting capabilities (cf. e.g. [HLN99]). Could a restricted use of the ordering of the database elements be used to overcome some of these limitations? A possible way to reconcile the desire to exploit the order with the need to ensure data independence, is to consider only so called *order-invariant formulae* [Gur88]. A formula which uses a binary relation symbol $<$ is order-invariant[2], if, whenever it is satisfied in a finite structure with respect to one total order which is chosen as an interpretation for $<$, then it is satisfied for *every* interpretation of $<$ by a total order of the structure as well. This definition is not very practical, since order-invariance of formulae cannot be checked algorithmically [Gur88]. That is, the user formulating a query has to take care that his or her formula is indeed order-invariant. However, it turns out that this very restricted use of the order indeed allows FO-formulae to express queries which cannot be expressed without the help of an order (as noted first by Gurevich, see [EF99] or [Lib04]). This leads to a plethora of questions. Some examples of questions which are investigated in this thesis follow.

What are the limitations of the expressive power of order-invariant formulae?
For instance, one concrete notion of locality which applies to queries defined by FO is called *Gaifman locality* [HLN99]. A (unary) query is Gaifman-local if there is a constant r, called the *locality radius* of the query, such that the query does not distinguish between any two elements of a structure which have isomorphic neighbourhoods of radius r (i.e. the substructures induced by all elements at distance r to the elements are isomorphic). It is known that queries which are definable by order-invariant formulae are also Gaifman local [GS00].

Does there exist an *effective syntax* for order-invariantly FO-definable queries?
That is, does there exist a logic whose set of formulae is decidable and which expresses exactly the same queries as order-invariant FO-formulae? Answering this question in full generality seems quite forbidding. Are there restricted classes of structures for which it can be answered? For instance, it has been shown that FO itself is an effective syntax for order-invariant first-order logic on various kinds of trees [BS09b], i.e. the invariant use of the order does not add anything to the expressive power of FO on trees.

How succinct are order-invariant definitions of queries?
If a fitting effective syntax is found for order-invariant FO on some restricted class of structures, what happens to the size of order-invariant FO-formulae when they are translated to this syntax? More generally, it can be asked for any family of queries that is both expressible in FO with an order and in (some extension of) FO without order if the use of an order enables a definition which is *more succinct* (i.e. which uses shorter formulae) than a definition without an order.

Given the fundamental nature of FO as a query language, it seems highly desirable to gain a better understanding of these topics.

[2] Precise definitions follow in the main part of the thesis

Descriptive complexity. As a remedy against the expressive weakness of FO as a query language, Aho and Ullman introduced a query language with a form of recursion. A corresponding extension of FO is known as *least fixed-point logic* (LFP). A seminal result of Immerman [Imm86] and Vardi [Var82] states that LFP *captures* polynomial time on ordered finite structures. That is, a query to structures with a total order can be expressed in LFP if, and only if, it can be computed in deterministic polynomial time. The Immerman-Vardi theorem belongs to *descriptive complexity theory* (cf. e.g. [Imm99],[EF99],[Lib04]) which investigates the connection between the computational complexity of queries to the complexity of logical formulae which express the queries. Ultimately, the goal is to characterise complexity classes on all finite structures, with and without an order, by logics as achieved by the theorem of Fagin [Fag74] which states that existential second-order logic captures non-deterministic polynomial time on finite structures. In fact, the Immerman–Vardi–theorem implies that the order-invariant formulae of least fixed-point logic capture deterministic polynomial time on finite structures, but this is unsatisfactory since we do not have an effective syntax for the order-invariant least fixed-point formulae. The question if there exists such an effective syntax is considered the major open problem of descriptive complexity theory (cf. e.g. [Gro08]).

While order-invariant FO seems to be too weak to capture any of the common complexity classes studied in complexity theory, the concept of order-invariance can be extended to obtain a capturing result. Specifying a total order on a structure with n elements is equivalent to assuming that the universe is an initial segment $0, \ldots, n-1$ of the natural numbers. This way, the order allows to transfer any relation or operation on the natural numbers such as the natural linear order $<$, addition $+$, or multiplication $*$ to a relation on the structure. We call these relations *numerical predicates*. Formulae can then be granted access to a set $\mathcal{N}$ of numerical predicates and the notion of order-invariance can naturally be extended to a notion of $\mathcal{N}$-invariance. In particular, we can consider the set of *all* numerical predicates, which we denoted by $\mathcal{ARB}$. A result of Immerman [Imm87] implies that $\mathcal{ARB}$-invariant FO-formulae capture the complexity class non-uniform AC^0. That is, a query is definable by an $\mathcal{ARB}$-invariant FO-formula iff it can be computed by a non-uniform family of circuits of bounded depth, unbounded fan-in, and polynomial size. In such a circuit family, there is a different circuit for each input size and the non-uniformity of the circuit family means that no restrictions are imposed on the way the circuits are obtained. This mirrors the unrestricted use of numerical predicates on the logic side. The non-uniformity of AC^0 does not allow for a meaningful comparison of AC^0 to complexity classes defined by Turing machines since it introduces even uncomputable problems into the class AC^0. For this reason, uniformity conditions which put computational limitations on the construction of the circuit for any given input size in a circuit family have been studied. It is a remarkable fact that natural uniformity conditions correspond to natural restrictions of numerical predicates. Barrington, Immerman, and Straubing [BIS90] established a result which implies that $\{<, +, *\}$-invariant FO-formulae capture DLOGTIME-uniform AC^0. This leads to analogous questions as for order-invariant least fixed point formulae. That is, is there an effective syntax for the $\{<, +, *\}$-invariant FO-formulae? Can we find answers to this question if we apply stronger restrictions to the numerical predicates or if we restrict the class of considered structures? It is known that $\mathcal{ARB}$-invariant FO-formulae share some of the limitations of FO. For instance, they are unable to count, i.e. such formulae are unable to express that the size of some unary relation is even [Ajt83]. Furthermore, $\mathcal{ARB}$-invariant FO-formulae are also local in a certain precise sense [AvMSS12]. This leads to the question what happens if we combine $\mathcal{ARB}$-invariance with limited forms of counting. Is the resulting class of formulae still local?

1.1 Contributions

This section presents a short summary of the main results of this thesis which can be divided into three parts, each corresponding to a chapter. More detailed discussions of the results are given at the beginning of the corresponding chapters.

$\mathcal{ARB}$-invariant first-order logic with modulo-counting

We study Gaifman locality and Hanf locality of $\mathcal{ARB}$-invariant formulae of FO with modulo-p-counting quantifiers (FO+MOD$_p$). We draw a detailed picture of locality and non-locality properties of such formulae. For example, on the class of all finite structures, for any $p \geqslant 2$, $\mathcal{ARB}$-invariant FO+MOD$_p$-formulae are neither Hanf nor Gaifman local with respect to a sublinear locality radius. However, if p is an odd prime power, $\mathcal{ARB}$-invariant FO+MOD$_p$-formulae are *weakly* Gaifman local with a polylogarithmic locality radius. And when restricting attention to the class of finite words, for odd prime powers p, $\mathcal{ARB}$-invariant FO+MOD$_p$-formulae are both Hanf local and Gaifman local with a polylogarithmic locality radius. Our negative results extend examples of order-invariant FO+MOD$_p$ formulae presented in Niemistö's PhD thesis [Nie07]. Our positive results can be seen as extension of the results of [AvMSS12], which considered Gaifman locality of $\mathcal{ARB}$-invariant FO-formulae without counting quantifiers, and of results of [Nie07] which considered locality of $\{<\}$-invariant FO+MOD$_p$-formulae. These results build on the close connection between $\mathcal{ARB}$-invariant formulae and Boolean circuits.

These results have been obtained in collaboration with Nicole Schweikardt. They have been first announced in the conference paper [HS13] and later published in full detail in the journal paper [HS16] on which the presentation in this thesis is based.

Addition-invariant first-order logic and regular tree languages

We consider regular tree languages of ranked trees which are represented as successor structures. We show that there is an effective syntax for the regular tree languages which are definable by $\{+\}$-invariant (addition-invariant) FO-formulae. More concretely, we show that the regular tree languages which are definable by addition-invariant FO-formulae are exactly the tree languages which are definable by FO with cardinality predicates (FO$_{\text{card}}$) which allow to express, for each positive integer m, that the number of elements of a structure is divisible by m. To this end, we obtain an algebraic characterisation of the FO$_{\text{card}}$-definable tree languages. We show that this characterisation is decidable. That is, we obtain an algorithm which, on input of a tree automaton, decides if the language recognised by this automaton is $\mathrm{FO}_{\mathrm{card}}$-definable (and hence, by our characterisation, definable by addition-invariant FO-formulae). The algebraic characterisation of the $\mathrm{FO}_{\mathrm{card}}$-definable tree languages can be seen as an extension of the characterisation of the FO-definable tree languages of Benedikt and Segoufin [BS09a]. For the proof of our characterisation, we gain important insights into the algebraic operations introduced by [BS09a] which help to clarify several previous results. Our characterisation of the addition-invariantly definable languages extends results of Schweikardt and Segoufin [SS10] from words to trees. This extension rests on a different approach than the proof of [SS10] which contains a significant gap.[3]

These results have been obtained in collaboration with Nicole Schweikardt. They have been announced in the conference paper [HS12], but the full results of this thesis have not been published before.

Order-invariant first-order and monadic second-order logic and structures of bounded tree-depth

We study the expressive power of order-invariant FO-sentences and order-invariant sentences of monadic second order logic (MSO) on classes of structures of bounded tree-depth. We show that on structures of bounded tree-depth, plain FO provides an effective syntax for order-invariant FO. Here we say that a formula is *plain* if it does not use a linear order. Furthermore, we show that, analogously, plain FO+MOD provides an effective syntax for order-invariant MSO on bounded tree-depth structures. We motivate these results by showing that order-invariance remains undecidable for coloured graphs of tree-depth at most 2, i.e. vertex-coloured star graphs. Our proof approach allows for giving upper bounds on the *succinctness* of order-invariant FO compared to plain FO on structures of bounded tree-depth. That is, we show that, on classes of bounded tree-depth structures, each order-invariant FO-sentence is equivalent to a plain FO-sentence whose size is at most d-fold exponential in the quantifier-rank of the order-invariant sentence, where d is the bound on the tree-depth of the considered structures. This proof approach can also be used to obtain a new proof of a known result of [EGT12] that plain MSO has the same expressive power as plain FO on structures of tree-depth at most d. The simplification achieved in the present proof allows to obtain analogous upper bounds as in the previous result, i.e. the size of the FO-sentence is at most d-fold exponential in the quantifier-rank of the MSO-sentence. This is complemented by a lower bound which shows that this result is essentially optimal.

These results have been obtained in collaboration with Kord Eickmeyer and Michael Elberfeld. They have been announced in the conference paper [EEH14] and an extended

[3] In the proof of [SS10, Theorem 3.6], it is stated that "(...) It thus suffices to consider prime counters of size $< N$". This claim seems to be unjustified and there seems to be no easy fix for this problem. This gap was acknowledged by the authors of [SS10]. In private communication with the author of this thesis, Luc Segoufin has outlined a different proof of [SS10, Theorem 3.6] which, however, rests on recent results on logic on words for which no extension to trees is known.

version has been made available on the Internet [EEH16]. The presentation in this thesis is based on the latter version.

For all results of this thesis which have been obtained in collaboration, the author has obtained permission from his coauthors to use the results for this thesis.

1.2 Outline

Chapter 2 recalls the fundamental concepts and introduces the necessary notation that is used throughout the thesis. It is followed by three chapters which contain the main results. Chapter 3 contains the results about $\mathcal{ARB}$-invariant FO+MOD, Chapter 4 covers addition-invariant FO on trees, and Chapter 5 is concerned with the results on structures of bounded tree-depth. After these three chapters, Chapter 6 concludes the thesis with some final remarks. In particular, the most interesting open questions are recollected in this last chapter.

The three main chapters (chapters 3–5) can be read independently. Each chapter starts with an informal introduction to the topics of the chapter, followed by a discussion which relates the contributions of the chapter to known results in the literature. After this comes a section which, more formally, introduces the necessary concepts and the notation needed throughout the chapter. This is followed by the main part of the chapter. We conclude each chapter by discussing possibilities for further research.

Preliminaries

We assume that the reader is familiar with the fundamentals of mathematical logic in general (cf. [EFT07]) and finite model theory in particular (cf. [Imm99], [Lib04], [EF99]). We also assume some familiarity with the fundamentals of formal languages, computability (cf. [HU79]), and computational complexity (cf. [AB09]). In this chapter, we recall some concepts and introduce the notation that will be used throughout this thesis. A reader aquainted with these subjects can safely skip these preliminaries and come back here if some notation is not clear. More specific notation is also introduced at the beginning of each of the main chapters.

2.1 General mathematical notation

Sets.

We write $A \setminus B$ for the *complement* of B in A. By 2^A we denote the *power set* of A, i.e. the set $\{Y : Y \subseteq A\}$. We write A^n for the *set of all sequences*/tuples of length $n \in \mathbb{N}$ with elements from A, we write A^* for the union of A^n set of all sequences of finite length and A^+ for this set without the empty sequence. The *empty sequence* is denoted by $()$.

Relations.

A relation is a *partial order* if it is reflexive, transitive, and antisymmetric. A *strict partial order* is a relation which is irreflexive,transitive, and asymmetric. A *linear order* is a partial order which is total and a *strict linear order* is a strict partial order which is total. The *index* of an element x in a linear order (or a strict linear order) $<$ on a set M is $|\{y \in M : y < x, y \neq x\}|$. We denote the index of x by $\mathrm{ind}_<(x)$. Then $\mathrm{ind}_< : M \to \{0, \dots, |M| - 1\}$ is a bijection.

The *restriction* of a k-ary relation R on a set M to a subset $N \subseteq M$ is the relation $R{\restriction}N := R \cap N^k$.

Numbers.

The sets of natural numbers with and without 0 are denoted by $\mathbb{N}$ and $\mathbb{N}^+$, respectively. For $n, m \in \mathbb{N}$ such that $n \leqslant m$, we define

$$[n, m] \;:=\; \{n, n+1 \dots, m\}.$$

If $n > 0$, we let

$$[n] \ := \ [0, n-1]$$

denote the *initial segment* of the first n natural numbers.

For integers i, j, p with $p \geqslant 2$, we write $i \equiv j \ (mod \, p)$ (and say that i is congruent j modulo p) iff there exists an integer k such that $i = j + kp$. By $x \operatorname{rem} m$ we denote the non-negative remainder when dividing x by m. We write $\operatorname{lcm} S$ and $\gcd S$ to denote the *least common multiple* and the *greatest common divisor* of the elements in a finite set $S \subseteq \mathbb{N}^+$. We write $\log n$ to denote the logarithm of a number n with respect to base 2, and we often simply write $\log n$ instead of $\lfloor \log n \rfloor$.

Words

For us, an *alphabet* is a finite set Σ, a *word over* Σ is a finite sequence with elements from Σ, and a *language* is a set of words. We write ϵ for the *empty word*. We write $|w|$ for the *length* of a word $w \in \Sigma^+$. For each $a \in \Sigma$, we write $|w|_a$ for the number of occurrences of the letter a in the word w. A word $v \in \Sigma^*$ is a *factor* of another word w if $w = u_1 v u_2$ for words $u_1, u_2 \in \Sigma^*$. If $u_1 = \epsilon$, then v is *prefix* of w. If, furthermore, $u_2 \neq \epsilon$, then v is a *proper prefix*. We write $u \trianglelefteq v$ if u is a prefix of v and we write $u \triangleleft v$ if u is a proper prefix of v. The relations $\trianglelefteq$ and $\triangleleft$ are a partial and a strict partial order, respectively. We say that two words u, v are *comparable* if either $u \trianglelefteq v$ or $v \trianglelefteq u$; otherwise, u, v are *incomparable* which we denote by $u \parallel v$. A language L is *prefix closed* if $v \in L$ and $u \trianglelefteq v$ implies $u \in L$.

For each language L and each word w, we let $wL := \{wu : u \in L\}$.

2.2 Logic

In this thesis, we consider *first-order (predicate) logic*, which we abbreviate as FO, and extensions of first-order logic which allow for the use of numerical predicates or different quantifiers. For a precise definition of the syntax and semantics of FO, see e.g. [EFT07]. The notation and concepts that we use are closest to those described in the books [Lib04], [EF99].

Structures

For us, a *signature* σ is a set of *relation and constant symbols*. (We never use function symbols.) Each relation symbol R has an *arity* $\operatorname{ar}(R) \in \mathbb{N}^+$. Sometimes it is more convenient to treat signatures as sequences; for instance, for signatures σ, τ, we often write σ, τ to denote the signature $\sigma \cup \tau$.

A *structure* is a tuple $\mathfrak{A} := (A, R_1^{\mathfrak{A}}, \ldots, R_\ell^{\mathfrak{A}}, c_1^{\mathfrak{A}}, \ldots, c_m^{\mathfrak{A}})$ where A is a non-empty set (the *universe* of $\mathfrak{A}$), $R_1, \ldots, R_\ell$ and $c_1, \ldots, c_m$ are relation and constant symbols, for some $\ell, m \in \mathbb{N}$, $R_1^{\mathfrak{A}}, \ldots, R_\ell^{\mathfrak{A}}$ are relations of arity $\operatorname{ar}(R_1), \ldots, \operatorname{ar}(R_\ell)$ on the set A, and $c_1^{\mathfrak{A}}, \ldots, c_m^{\mathfrak{A}}$ are elements of A. The set $\sigma := \{R_1, \ldots, R_\ell, c_1, \ldots, c_m\}$ is the *signature of* $\mathfrak{A}$ and we say that $\mathfrak{A}$ is a *σ-structure*.

For each $\tau \subseteq \sigma$, the *τ-reduct* of $\mathfrak{A}$, which we denote by $\mathfrak{A} \upharpoonright \tau$, is the structure obtained by omitting all relations and constants which do not correspond to symbols from τ. If $\mathfrak{B} = \mathfrak{A} \upharpoonright \tau$ for some $\tau \subseteq \sigma$, then $\mathfrak{A}$ is a *σ-expansion* of $\mathfrak{B}$. Each subset B of the universe of $\mathfrak{A}$ induces a *substructure* $\mathfrak{A} \upharpoonright B$ of $\mathfrak{A}$. This is the σ-structure with the universe B whose relations are obtained by restricting the relations of $\mathfrak{A}$ to B.

We use the following important conventions regarding structures and signatures.

Proviso 1 (Naming of structures)**.** *Structures are denoted by Fraktur letters like $\mathfrak{A}, \mathfrak{B}, \mathfrak{C}, \ldots$ and their universes by the corresponding Latin letters $A, B, C, \ldots$.*

A signature is *relational* if it does not contain constant symbols and a *relational structure* is a structure over a relational signature.

Proviso 2 (Signatures are relational)**.** *If we say "signature", we mean "relational signature".*

Proviso 3 (Structures are finite and relational)**.** *If we say "structure", we mean "finite relational structure".*

There are a few exceptions to these rules but those are clearly recognisable. For instance, for graphs and words we often use names like G and w although we consider them as structures. If we explicitly define a structure with universe $\mathbb{N}$, then it is clearly not finite. In the rare case where we need constant symbols, we state this explicitly.

Classes of structures and queries

A class $\mathfrak{C}$ of structures is *closed under isomorphism* if $\mathfrak{A} \in \mathfrak{C}$ and $\mathfrak{B} \cong \mathfrak{A}$ implies $\mathfrak{B} \in \mathfrak{C}$, for all structures $\mathfrak{A}, \mathfrak{B}$. The *isomorphism closure* of $\mathfrak{C}$ is the inclusion-minimal class which contains $\mathfrak{C}$ and which is closed under isomorphism.

Proviso 4 (Closure under isomorphism)**.** *We assume that all* classes $\mathfrak{C}$ *of structures are closed under isomorphism, i.e. if we speak of $\mathfrak{C}$, we actually mean its isomorphism closure.*

A *k-ary query* q on a class of structure $\mathfrak{C}$ is a mapping which takes every structure $\mathfrak{A} \in \mathfrak{C}$ to a k-ary relation $q(\mathfrak{A}) \subseteq A^k$ which is invariant under isomorphisms. That is, if π is an isomorphism from $\mathfrak{A} \in \mathfrak{C}$ to $\mathfrak{B} \in \mathfrak{C}$, then for all $\overline{a} = (a_1, \ldots, a_k) \in A^k$ we have $\overline{a} \in q(\mathfrak{A})$ iff $\pi(\overline{a}) = (\pi(a_1), \ldots, \pi(a_k)) \in q(\mathfrak{B})$. A 0-ary query is also called *Boolean*. We identify each Boolean query q on a class $\mathfrak{C}$ with the class of structures $\mathfrak{C}_q \subseteq \mathfrak{C}$ such that

$$\mathfrak{A} \in \mathfrak{C}_q \iff () \in q(\mathfrak{A}),$$

for each $\mathfrak{A} \in \mathfrak{C}$. Analogously, each class of structures $\mathfrak{D} \subseteq \mathfrak{C}$ is identified with a Boolean query $q_{\mathfrak{D}}$ on $\mathfrak{C}$.

For each signature σ, we let

$$\text{FIN}_\sigma := \{\mathfrak{A} : \mathfrak{A} \text{ is a } \sigma\text{-structure}\}$$

and we let

$$\text{FIN} := \bigcup_\sigma \text{FIN}_\sigma.$$

Special structures: graphs and words

For us, a *(directed) graph* is a structure over the signature $\{E\}$, where E is a binary relation symbol. In the context of graphs, we use the usual graph theoretic notation. In particular, we use names like G, H for graphs, the elements of a graph are called *vertices*, and the relation of a graph is its *edge set*. We consider *undirected graphs* as directed graphs whose edge relation is symmetric and irreflexive. When discussing undirected graphs, we consider the edges (u, v) and (v, u) as equal.

Let Σ be a finite set and consider the signature $\sigma_\Sigma := \{E\} \cup \{P_a : a \in \Sigma\}$ where each P_a is a unary relation symbol. A Σ*-coloured graph* is a σ_Σ-expansion $\mathfrak{A}$ of a graph where the sets $(P_a^{\mathfrak{A}})_{a\in\Sigma}$ partition the vertex set, i.e. they are pairwise disjoint and their union is the whole vertex set.

We associate each word $w = w_0 \cdots w_{n-1} \in \Sigma^n$ with a Σ-coloured graph $\mathfrak{S}_w$ over the vertex set $[n]$ where the unary relation corresponding to the symbol $a \in \Sigma$ consists of all positions $i \in [n]$ such that $w_i = a$, and the edge relation is the natural *successor relation* on $[n]$ which contains an edge $(i, i+1)$ for each $i \in [n-1]$. Structures which are isomorphic to a structure $\mathfrak{S}_w$, for some word $w \in \Sigma^+$, are called Σ*-word structures*.

Proviso 5 (Word structures)**.** *We identify each word with its associated word structure $\mathfrak{S}_w$. The set Σ^+ is consequently identified with the class of all Σ-word structures.*

Note that in contrast to parts of the literature, our representation for words does not contain the natural linear order on a word as a relation.

Gaifman graphs and distance in structures

We transfer graph theoretic notions from graphs to general structures via the notion of Gaifman graphs of structures. The *Gaifman graph* $\mathfrak{G}(\mathfrak{A})$ of a structure $\mathfrak{A}$ is the undirected graph with vertex set A containing an edge between $x, y \in A$ iff $x \neq y$ and x and y occur together in a tuple in one of the relations of $\mathfrak{A}$. A structure $\mathfrak{A}$ is *connected* if $\mathfrak{G}(\mathfrak{A})$ is connected and a (connected) *component* of $\mathfrak{A}$ is a substructure of $\mathfrak{A}$ which is induced by a connected component of $\mathfrak{G}(\mathfrak{A})$. The *distance* $\text{dist}^{\mathfrak{A}}(a, b)$ between elements a, b of $\mathfrak{A}$ is their distance in $\mathfrak{G}(\mathfrak{A})$. That is, the length of a shortest path between a and b in $\mathfrak{G}(\mathfrak{A})$, if a, b belong to the same component, and ∞ otherwise. More generally, for tuples $\bar{a} := (a_1, \ldots, a_\ell)$, $\bar{b} := (b_1, \ldots, b_m)$ of elements of $\mathfrak{A}$, we let $\text{dist}^{\mathfrak{A}}(\bar{a}, \bar{b}) := \min_{i\in[1,\ell], j\in[1,m]} \text{dist}^{\mathfrak{A}}(a_i, b_j)$. The k*-neighbourhood* $N_k^{\mathfrak{A}}(\bar{a})$ of $\bar{a}$ in $\mathfrak{A}$ is the set of all elements b of $\mathfrak{A}$ with $\text{dist}^{\mathfrak{A}}(b, \bar{a}) \leqslant k$. The k*-sphere* around $\bar{a}$ in $\mathfrak{A}$ is the substructure $\mathcal{N}_k^{\mathfrak{A}}(\bar{a})$ of $(\mathfrak{A}, \bar{a})$ induced by the set $N_k^{\mathfrak{A}}(\bar{a})$. We say that $\mathcal{N}_k^{\mathfrak{A}}(\bar{a})$ and $\mathcal{N}_k^{\mathfrak{A}}(\bar{b})$ are *disjoint* if $N_k^{\mathfrak{A}}(\bar{a})$ and $N_k^{\mathfrak{A}}(\bar{b})$ are disjoint.

First-order logic and definability

We write FO[σ] to denote the set of all first-order formulae over the signature σ and FO to denote the set of all first-order formulae. We usually omit braces around the sets in this notation, i.e. we write FO[E] instead of FO[$\{E\}$]. For a formula $\varphi \in$ FO, we write *free*(φ) to denote the free variables of φ. We write $\varphi(x_1, \ldots, x_k)$ to indicate that *free*(φ) $= \{x_1, \ldots, x_k\}$. A *sentence* is a formula without free variables. For a first-order sentence φ and a structure $\mathfrak{A}$, we write $\mathfrak{A} \models \varphi$ to denote that $\mathfrak{A}$ satisfies φ. For a formula $\varphi(x_1, \ldots, x_k)$ and $a_1, \ldots, a_k \in A^k$, we write $\mathfrak{A} \models \varphi[a_1, \ldots, a_k]$ if φ is satisfied in $\mathfrak{A}$ when interpreting the free occurrences of the variables $x_1, \ldots, x_k$ with the elements $a_1, \ldots, a_k$. On each σ-structure $\mathfrak{A}$, an FO[σ]-formula $\varphi(\overline{x})$ with k free first-order variables defines a k-ary relation

$$\varphi(\mathfrak{A}) \ := \ \{\overline{a} \in A^k : \mathfrak{A} \models \varphi[\overline{a}]\}.$$

We can hence consider φ as a Boolean query on the class of all structures. For each class $\mathfrak{C}$ of structures, $\varphi \restriction \mathfrak{C}$ denotes the restriction of this Boolean query to $\mathfrak{C}$. Given a query q on $\mathcal{C}$, we say that q *is* FO-*definable on* $\mathcal{C}$ if there is an FO-sentence such that $q = \varphi \restriction \mathfrak{C}$. We say that a class $\mathfrak{D} \subseteq \mathfrak{C}$ is FO-definable iff the associated query $q_{\mathfrak{D}}$ is FO-definable on $\mathfrak{C}$. In these definitions, if $\mathfrak{C}$ is the class FIN of all finite structures, we say that the query q or the class of structures $\mathfrak{D}$ is FO-*definable*.

Other logics

The previous definitions are analogously used for other logics. We write MSO for the set of all formulae of *monadic second-order logic*, i.e. the extension of FO by set quantifiers. We also consider the logic FO+MOD which is obtained from FO by allowing the use of *modulo-counting quantifiers* of the shape $\exists^{i \pmod p}$, for each $i \in \mathbb{N}$ and $p \in \mathbb{N}^+$, which express that i is the remainder of the division of the number of satisfying assignments of a formula by p. More precisely, the semantics of the quantifiers is defined by $(\mathfrak{A}, \bar{a}) \models \exists^{i \pmod p} x\, \varphi(x, \bar{y})$ iff $|\{b \in A : \mathfrak{A} \models \varphi(b, \bar{a})\}| \equiv i \pmod p$, where $\mathfrak{A}$ is a structure and $\bar{a}$ is a tuple of its elements. We write FO+MOD$_p$ for the set of formulae of FO+MOD where the only modulo-counting quantifiers allowed are those with modulus p. We also consider the extension of MSO by arbitrary modulo-counting quantifiers. In the literature, this logic is usually called CMSO. We stick to this convention instead of using a name which is more consistent wit hour naming of the corresponding extension of FO .

Complexity of formulae

Quantifier rank. The *quantifier rank* of a formula φ, i.e. the maximum number of quantifiers on any directed path of the syntax tree of φ, is denoted by $\mathrm{qr}(\varphi)$. This definition applies to all of the different kinds of logical formulae that we consider. If $\mathrm{qr}(\varphi) = 0$, then φ is *quantifier-free*. For any of the logics L that we consider and for each $q \in \mathbb{N}$, we write $\mathfrak{A} \equiv_q^{\mathrm{L}} \mathfrak{B}$ for $q \in \mathbb{N}$ if the structures $\mathfrak{A}$ and $\mathfrak{B}$ satisfy the same L-sentences of quantifier rank at most q. When it is obvious from the context which logic L we mean, we omit it from this notation.

Quantifier alternation. For each FO-formula φ, there is an equivalent formula in *prenex normal form*. That is, a formula of the form $Q_1 \ldots Q_\ell\ \psi$ where ψ is quantifier-free and $Q_1, \ldots, Q_\ell \in \{\exists, \forall\}$. For us, the *quantifier alternation depth* of φ, which we denote by $\mathrm{qad}(\varphi)$, is the minimum number of alternations between $\exists$ and $\forall$ in the word $Q_1 \cdots Q_\ell$ in all prenex normal form sentences which are equivalent to φ.

Size of formulae. A $\mathrm{FO}[\sigma]$-formula φ which uses variables $x_1, \ldots, x_k$ can be viewed as a word over the alphabet $\sigma \cup \{x_1, \ldots, x_k, =, \exists, \forall, \neg, \wedge, \vee, \rightarrow, \leftrightarrow, (,)\}$. The *length* or *size* of φ is the length of this word and is denoted by $|\varphi|$. For FO+MOD-formulae, we use an analogous definition, where we assume that the numbers occurring in modulo-counting quantifiers are encoded in binary. We extend these notations to sets Φ of formulae with the definitions $\|\Phi\| := \max_{\varphi \in \Phi} |\varphi|$ and $\mathrm{qad}(\Phi) := \max_{\varphi \in \Phi} \mathrm{qad}(\Phi)$.

Interpretations

Throughout this thesis, we make extensive use of *first-order interpretations*, which are a basic tool of mathematical logic (see e.g. [EF99]). We use only a simple special case of the general definition.

Definition 2.2.1 (Interpretation)**.** Let σ and τ be signatures. A (σ, τ)*-interpretation* is a partial mapping $\mathcal{I}$ from σ-structures to τ-structures $\mathcal{I}(\mathfrak{A})$. A (σ, τ)-interpretation $\mathcal{I}$ is *first-order definable* (or just *first-order*, for short) if there exist $\mathrm{FO}[\sigma]$-formulae $\varphi_{\mathrm{univ}}(x)$ and $\varphi_R(x_1, \ldots, x_\ell)$, for each relation symbol $R \in \tau$ of arity ℓ, such that for each σ-structure $\mathfrak{A}$ for which $\mathcal{I}(\mathfrak{A})$ is defined, the universe of $\mathcal{I}(\mathfrak{A})$ is

$$\mathcal{I}(A) \ = \{a \in A : \mathfrak{A} \models \varphi_{\mathrm{univ}}(a)\},$$

and each R is interpreted in $\mathcal{I}(\mathfrak{A})$ by the relation

$$R^{\mathcal{I}(\mathfrak{A})} \ = \{(a_1, \ldots, a_\ell) \in \mathcal{I}(A)^\ell : \mathfrak{A} \models \varphi_R(a_1, \ldots, a_\ell)\}.$$

The fundamental lemma about interpretations shows that a first-order definable (σ, τ)-interpretation can be used to convert each $\mathrm{FO}[\tau]$-sentence φ to a $\mathrm{FO}[\sigma]$-sentence $\mathcal{I}(\mathfrak{A})$ which simulates the evaluation of φ in the τ-structure $\mathcal{I}(\mathfrak{A})$ when evaluated in a σ-structure $\mathfrak{A}$.

Lemma 2.2.2 (Interpretation Lemma)**.** *Let σ and τ be signatures and let $\mathcal{I}$ be a first-order (σ, τ)-interpretation. Let $\mathrm{L} \in \{\mathrm{FO}, \mathrm{MSO}\}$. For each $\mathrm{L}[\tau]$-sentence φ, there is an $\mathrm{L}[\sigma]$-sentence $\mathcal{I}(\varphi)$ such that for all σ-structures $\mathfrak{A} \in \mathcal{C}$ for which $\mathcal{I}(\mathfrak{A})$ is defined,*

$$\mathfrak{A} \models \mathcal{I}(\varphi) \iff \mathcal{I}(\mathfrak{A}) \models \varphi.$$

The quantifier rank of $\mathcal{I}(\varphi)$ is at most $\mathrm{qr}(\varphi) + m$, where m is the maximum of the quantifier ranks of the formulae defining $\mathcal{I}$.

The proof is standard: rewrite φ according to the formulae defining $\mathcal{I}$, relativising quantifiers to φ_{univ} (e.g. $\exists x\ \psi$ becomes $\exists x\ \varphi_{\text{univ}}(x) \wedge \psi$) and replacing atomic formulae $R(x_1, \ldots, x_{\text{ar}(R)})$ by $\varphi_R(x_1, \ldots, x_{\text{ar}(R)})$.

In this thesis, we are only concerned with first-order interpretations. We often describe interpretations informally in such a way that it hopefully becomes easy for the reader to see that the interpretation is first-order definable. In fact, we often describe how one formula can be rewritten to obtain a desired formula, without appealing to the formal definition of an interpretation and the Interpretation Lemma. The correctness of these constructions can be verified using the Interpretation Lemma.

2.3 Numerical predicates

An r-ary *numerical predicate* is a relation $P \subseteq \mathbb{N}^r$. We write $\mathcal{ARB}$ for the set of all[1] numerical predicates. The following concrete numerical predicates are used in the results or in the discussion of this thesis:

- $<$: the natural linear order on $\mathbb{N}$.
- S : the successor relation on $\mathbb{N}$ (i.e. $(x, y) \in \mathrm{S}$ iff $y = x + 1$).
- $+$: the set of all tuples $(x, y, z) \in \mathbb{N}$ such that $x + y = z$.
- $*$: the set of all tuples $(x, y, z) \in \mathbb{N}$ such that $x \cdot y = z$.

We want to grant logical formulae access to some set $\mathcal{N} \subseteq \mathcal{ARB}$ of numerical predicates. To this end, we also consider each r-ary numerical predicate as an r-ary relation symbol. For instance, in formulae we write $<$ to mean a binary relation symbol. To avoid any ambiguities, we need the following proviso.

Proviso 6. *We assume that signatures denoted by lower-case Greek letters $\sigma, \tau, \ldots$ do not contain any relation symbols corresponding to numerical predicates.*

$\mathbb{N}$-embeddings

An $\mathbb{N}$-*embedding* (or just *embedding*) of a structure $\mathfrak{A}$ with n elements is a bijection of A and the initial segment $[n]$. We can consider each numerical predicate P as a relation on $[n]$ by restricting it appropriately. Instead of $([n], P \restriction [n])$ we will write $([n], P)$. Accordingly, we write $([n], \mathcal{N})$ for the expansion of $[n]$ by all numerical predicates from the set $\mathcal{N}$. A *numerical $\mathcal{N}$-expansion* of a σ-structure $\mathfrak{A}$ on n elements is a $(\sigma, \mathcal{N})$-expansion of $\mathfrak{A}$ such that for some $\mathbb{N}$-embedding ι we have $\iota : \mathfrak{A} \restriction \mathcal{N} \cong ([n], \mathcal{N})$. For a given embedding ι, we also write $\mathfrak{A}^\iota$ to denote a numerical $\mathcal{ARB}$-expansion of $\mathfrak{A}$ whose embedding is ι.

A *linear order on a structure* $\mathfrak{A}$ is a linear order $<^{\mathfrak{A}}$ on A. In this case, $(\mathfrak{A}, <^{\mathfrak{A}})$ is a numerical $<$-expansion of $\mathfrak{A}$. Analogously, we call $+^{\mathfrak{A}}$ an *addition relation* on $\mathfrak{A}$ if $(\mathfrak{A}, +^{\mathfrak{A}})$ is a numerical $+$-expansion of $\mathfrak{A}$.

[1] In other words, $\mathcal{ARB}$ is the set of "arbitrary" numerical predicates, which explains the name.

Words and numerical predicates

Recall that we have identified words of length n with coloured graphs over the universe $[n]$ where the edge relation is the successor relation on $[n]$. For each set $\mathcal{N}$ of numerical predicates and each word w, we call the numerical $\mathcal{N}$-expansion of w whose embedding is the identity on $[n]$ the *canonical $\mathcal{N}$-expansion* of w. An FO$[\sigma_\Sigma, \mathcal{N}]$-sentence φ *defines* the language of all Σ-words whose canonical $\mathcal{N}$-expansions are models of φ. When we discuss words, we usually omit the signature σ_Σ and write FO$[\mathcal{N}]$ for FO$[\sigma_\Sigma, \mathcal{N}]$. [2]

Often, the expressive power of a logic can be extended by allowing the use of certain numerical predicates. For example, it is well-known that the FO-definable languages are a strict subset of the FO$[<]$-definable languages: for instance, the language $a^*ba^*bba^*$ is FO$[<]$-definable, but not FO-definable. For another example, observe that the language $\{a^n b^n : n \in \mathbb{N}^+\}$ is FO$[+]$-definable, but not FO$[<]$-definable (it is not even regular).

$\mathcal{N}$-invariance

To allow the use of numerical predicates in structures whose universes are arbitrary finite sets, we introduce the following definitions.

Definition 2.3.1 ($\mathcal{N}$-invariance)**.** An FO$[\sigma, \mathcal{N}]$-formula $\varphi(\bar{x})$, where $\bar{x}$ is a tuple of k variables, is *$\mathcal{N}$-invariant* on a class of structures $\mathcal{C} \subseteq$ FIN if $\varphi(\overline{x})$ does not distinguish between $\mathcal{N}$-expansions of structures from $\mathcal{C}$. That is,

$$(\mathfrak{A}_1, \bar{a}) \models \varphi \iff (\mathfrak{A}_2, \bar{a}) \models \varphi.$$

for all $\mathfrak{A} \in \mathcal{C}$, all $\bar{a} \in A^k$, and all $\mathcal{N}$-expansions $\mathfrak{A}_1$ and $\mathfrak{A}_2$ of $\mathfrak{A}$.

If $\varphi(\bar{x})$ is $\mathcal{N}$-invariant on $\mathcal{C}$ and $\mathfrak{A} \in \mathcal{C}$, $\bar{a} \in A^k$, we define $(\mathfrak{A}, \bar{a}) \models \varphi$ with the meaning that $\mathfrak{A}' \models \varphi$ *for some* $\mathcal{N}$-expansion $\mathfrak{A}'$ of $\mathfrak{A}$. Note that Proviso 6 assures that this definition is unambiguous, since the signature of $\mathfrak{A}$ does not contain any numerical predicates from $\mathcal{N}$. Note further that, by $\mathcal{N}$-invariance, we could equivalently require that $(\mathfrak{A}', \bar{a}) \models \varphi$ *for each* $\mathcal{N}$-expansion $\mathfrak{A}'$ of $\mathfrak{A}$.

The set of all $\mathcal{N}$-invariant formulae is denoted by $\mathcal{N}$-inv-FO. If $\mathcal{N} = \{<\}$, we also write $<$-inv-FO and speak of *order-invariant* FO. For the case where $\mathcal{N} = \{+\}$, we also write $+$-inv-FO and speak of *addition-invariant* FO. Note that $<$ can be defined from $+$ and hence we allow the use of $<$ in $+$-inv-FO-formulae. The definitions above do not rely on any particularities of FO and can be applied to any other logic (e.g. MSO, FO+MOD) that occurs in this thesis.

[2] Note that our definition of numerical predicates is different from the definition that is often used in the literature on logic on words (cf. [Str94]). But it is equivalent for logical purposes, since each of our r-ary numerical predicates corresponds to an $(r+1)$-ary numerical predicate according to the definition of [Str94].

Locality of $\mathcal{ARB}$-invariant FO with modulo-counting

In this chapter, we consider the locality and non-locality properties of the invariant formulae of FO with arbitrary numerical predicates and with modulo-counting quantifiers ($\mathcal{ARB}$-inv-FO+MOD).

3.1 Introduction

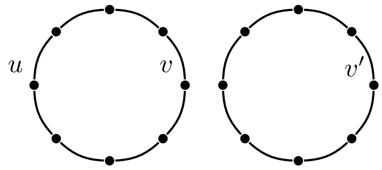
We start with an overview of the two main notions of locality. Afterwards, we discuss the contributions of this chapter.

3.1.1 Gaifman locality

The classical inexpressibility arguments for logics over finite structures often use back-and-forth systems or Ehrenfeucht-Fraïssé games (cf. e.g. [Lib04]). Often the application of these techniques involves nontrivial combinatorics. An alternative and intuitively appealing way for proving inexpressibility results for first-order formulae was introduced by Gaifman [Gai82]. He showed that each first-order formula is equivalent to a Boolean combination of formulae which speak only about certain neighbourhoods of a constant radius. Nowadays, Gaifman's result is often called *Gaifman's theorem* and the normal form provided by it is called a *Gaifman normal form*. As discussed with several illustrated examples in the paper of Gaifman, the existence of Gaifman normal forms for FO-formulae allows to prove non-expressibility results for FO. While Gaifman normal forms have a plethora of applications, many non-expressibility results can be derived from a weaker principle which was identified in [HLN99].[1] A k-ary query is called *Gaifman local with locality radius* $f(n)$ if, in a structure of sufficiently large cardinality n, the question whether a given tuple satisfies the query only depends on the isomorphism type of the tuple's $f(n)$-*sphere*, i.e. the substructure induced by the tuple's neighbourhood of radius $f(n)$. It is an immediate consequence of Gaifman's theorem that FO-definable queries are *constantly Gaifman local*, i.e. Gaifman local with a constant locality radius. The notion of Gaifman locality captures the essence of many non-expressibility results. It also has the strong advantage that it can be applied in settings where a generalisation of Gaifman's theorem is not necessarily possible.

[1]... and generalised to the present form with a non-constant locality radius in [AvMSS12].

Very often proofs based on Gaifman locality can be communicated in an intuitive visual way. While we have not given a precise definition of Gaifman locality yet, the present description should suffice to consider the following classical example which conveys the basic idea behind locality-based inexpressibility results. Consider the binary reachability query which assigns to a graph all pairs (u, v) of its vertices such that there is a path from u to v. Using Gaifman-locality, we can show that this query is not FO-definable. If the query were FO-definable, it would also be Gaifman local with a constant locality radius r. Consider a graph G consisting of two disjoint cycles of the same length where we have singled out vertices u, v, and v' as displayed below.



If the length of the cycles of G is chosen sufficiently large, then the r-spheres of the tuples (u, v) and (u, v') will be isomorphic, i.e. both are isomorphic to the disjoint union of paths of length $2r + 1$. By Gaifman locality, the reachability query should not distinguish between (u, v) and (u, v'). But, clearly, v is reachable from u and v' is not — a contradiction.

Note that, for the previous argument, we need only a weaker notion of locality which is defined like Gaifman locality, but the r-spheres of the considered tuples must be disjoint. This is called *weak Gaifman locality*.

Besides being local, another well-known limitation of FO is its inability to count. One approach to overcome this limitation considers FO with *unary counting quantifiers* (cf. [Lib04, Chapter 8]). A particular example are *modulo-counting* or *divisibility* quantifiers which allow to state that the number of satisfying assignments to a free variable of a formula is divisible by p, for some positive integer p. More generally, a unary counting quantifier states that the number of satisfying assignments to a free variable which occurs in a formula belongs to a fixed set N of natural numbers. For instance, for the modulo-p-counting quantifier, we have $N := \{0, p, 2p, \ldots\}$. Recall from Section 2.2 that we denote the extension of FO by the modulo-p-counting quantifier by FO+MOD$_p$ and the extension by all modulo-counting quantifiers by FO+MOD. It is known that even the FO+COUNT, the extension of FO by arbitrary unary counting quantifiers, defines only queries which are Gaifman-local with a constant locality radius [HLN99]. This result does not extend to $<$-inv-FO+COUNT, i.e. there are order-invariant FO+COUNT-formulae which are not constantly Gaifman local [LW02]. In contrast, it is known from results of Grohe and Schwentick [GS00] that all $<$-inv-FO-definable queries are constantly Gaifman local. A considerable strengthening of the result of [LW02] which was proved in the doctoral thesis of Niemistö [Nie05] shows that $<$-inv-FO+MOD, a seemingly weak fragment of $<$-inv-FO+COUNT, defines queries which are not constantly Gaifman local. In his thesis, Niemistö went on and showed that for each odd number p, all queries definable by $<$-inv-FO+MOD$_p$-formulae are weakly Gaifman local with a constant locality radius. Furthermore, he showed that for all p, $<$-inv-FO+MOD$_p$-definable queries satisfy a different notion of locality which he called *alternating locality*.

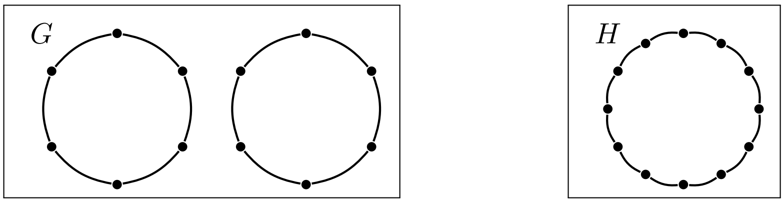
The result of Grohe and Schwentick leads to the natural question if queries which are FO-definable with the help of stronger sets of numerical predicates are also Gaifman local. A result of Hella (see [DLM07] for a proof) shows that constant Gaifman locality fails for $\{+, *\}$-inv-FO. It is an open problem if $+$-inv-FO is constantly Gaifman local. Grohe and Schwentick [GS00] also pointed out that it could be worthwhile to study Gaifman locality with a non-constant locality radius. This was taken up by Anderson, van Melkebeek, Schweikardt, and Segoufin [AvMSS12] who showed that each $\mathcal{ARB}$-inv-FO-definable query is Gaifman local with locality radius $\log(n)^c$, for some constant c. Building on the result of Hella and [DLM07], they also show that this locality radius is optimal. The upper bound on the locality radius of [AvMSS12] rests on the close connection between $\mathcal{ARB}$-inv-FO and circuit complexity (which we have discussed in the introduction of this thesis) and the strong lower bounds known for the parity language (i.e. the language of words over the alphabet $\{0, 1\}$ containing an even number of 1s) and related languages [Hås87].

3.1.2 Hanf locality

In a lemma, Hanf [Han65] showed that for each first-order sentence, there is a constant r such that two structures, finite or not, agree on the sentence provided that each isomorphism type of r-spheres occurs the same number of times in both structures or the number of occurrences is infinite in both structures. The last part of this condition concerning an infinite number of occurrences of sphere-types is obviously not very useful in the context of finite structures and it turns out that "infinite" can be replaced by "sufficiently large" [FSV95], i.e. larger than some constant t. This variant of Hanf's result is, in the context of finite model theory, nowadays often referred to as *Hanf's theorem* used to prove non-expressibility results. In the case of classes of structures of bounded degree, where the number of isomorphism types of r-spheres is finite, Hanf's theorem yields a normal form for first-order sentences, i.e. each sentence is equivalent to a Boolean combination of sentences which state that there exist at least t realisations of some isomorphism type of r-spheres. This has been called a *Hanf normal form* in [BK12] who investigated the complexity of constructing these normal form. It is known that similar normal forms exist for FO+MOD [Nur96] and that the modulo-counting quantifiers are, in some sense, the maximal set of counting quantifiers with this property [HKS16].

In the same way as the notion of Gaifman locality captures the useful essence of Gaifman's theorem for many non-expressibility proofs and allows for generalisation to logics beyond FO, the condition from Hanf's theorem also yields a useful general notion of locality of queries which is called *Hanf locality* [HLN99]. A Boolean query is *Hanf local with locality radius* $f(n)$ if the query does not distinguish between structures of sufficiently large cardinality, provided that the structures contain exactly the same number of occurrences of each isomorphism type of $f(n)$-spheres. Hanf's theorem immediately implies that all FO-definable queries are Hanf local with a constant locality radius. It is known that this result extends to FO+COUNT [HLN99].

As an example for the use of Hanf locality, we show that the graph connectivity query is not FO-definable. This is a well-known example. If graph connectivity were definable, it would be Hanf local with a constant locality radius r. Consider a graph G consisting of two disjoint cycles on $2r + 2$ vertices each and a graph H consisting of one disjoint cycle with $4r + 4$ vertices.



It is easy to verify that G and H contain the same number of occurrences of each isomorphism type of r-spheres (each such r-sphere in G and H is isomorphic to a path of length $2r$). By Hanf locality, G and H are not distinguished by the connectivity-query, but clearly H is connected and G is not — a contradiction.

For extensions of FO by numerical predicates, much less is known about Hanf locality than about Gaifman locality. For instance, it is an open question whether $<$-inv-FO is Hanf local with a constant or even a sublinear locality radius. However, it was shown in [AvMSS12] that each $\mathcal{ARB}$-inv-FO-definable query on words is Hanf local with locality radius $\log(n)^c$, for some constant c depending on the query.

3.2 Contributions

We discuss the main results of this chapter and their relation to previous results in the literature. The results of this chapter have been published in the paper [HS16]. The presentation is closely based on this paper and significant parts of this chapter agree with the content of the paper.

Our results give a detailed picture of the locality and non-locality properties of $\mathcal{ARB}$-invariant FO+MOD$_p$.

Shift-locality

We generalise weak Gaifman locality and the notion of alternating locality which was introduced by Niemistö [Nie07] to non-constant locality radii and unify these notions in a single notion of locality which we call shift locality. Our *first main result* (Theorem 3.5.7) establishes that, for prime powers p, $\mathcal{ARB}$-inv-FO+MOD$_p$ is shift local. This implies that, for even prime powers p, it is alternatingly local, and for odd prime powers p, it is weakly Gaifman local, with a polylogarithmic locality radius. We adapt the method introduced in [AvMSS12] where it was shown that $\mathcal{ARB}$-inv-FO-definable queries are Gaifman local with polylogarithmic locality using circuit complexity lower-bounds due to [Hås87]. We build on the well-known lower-bounds of [Smo87] for circuits with modular gates. Generalising our result from prime powers p to arbitrary numbers p can be expected to be difficult, since it would solve long-standing open questions in circuit complexity (see Remark 3.5.14). We give several examples which show that shift locality can be used in a similar way to Gaifman locality for proving non-expressibility results.

Locality lower-bounds

We adapt the examples of Niemistö [Nie07] to derive strong lower-bounds on the weak Gaifman locality of $\mathcal{ARB}$-inv-FO+MOD$_p$: For every natural number $p \geqslant 2$, order-invariant FO+MOD$_p$ is neither Hanf nor Gaifman local with a sublinear locality radius (see Section 3.6 and Proposition 3.5.2). For *even* numbers $p \geqslant 2$, order-invariant FO+MOD$_p$ is not even *weakly* Gaifman local with a sublinear locality radius (Proposition 3.5.4).

Locality on words

When restricting attention to the class of *word structures*, we obtain for odd prime powers p that $\mathcal{ARB}$-invariant FO+MOD$_p$ is both Hanf and Gaifman local with a polylogarithmic locality radius (Theorem 3.6.5 and Corollary 3.6.6). On the other hand, for even numbers $p \geqslant 2$, order-invariant FO+MOD$_p$ on word structures is neither Gaifman nor Hanf local with a sublinear locality radius (Proposition 3.5.4 and Section 3.6). This implies that order-invariant FO+MOD$_p$ defines languages which are not FO+MOD-definable, which shows that the conjecture of Benedikt and Segoufin [BS09b] that $<$-inv-FO+MOD and FO+MOD have the same expressive power on trees already fails on words.

3.3 Preliminaries

Numbers A number p is called a *prime power* if $p = \hat{p}^i$ for a prime $\hat{p}$ and an integer $i \geqslant 1$, and p is called an *odd* prime power if p's prime factor is different from 2 (i.e., p is odd).

Modulo-Counting quantifiers Note that, if m is a multiple of p, then FO+MOD$_m$ can express modulo p counting quantifiers, i.e.

$$\exists^{i \bmod p} x\, \varphi(x, \overline{y}) \;\equiv\; \bigvee_{j \in [m/p]} \exists^{jp+i \bmod m} x\; \varphi(x, \overline{y}).$$

3.4 Examples of $\mathcal{ARB}$-inv-FO+MOD-definable queries

We present two examples of $<$-inv-FO+MOD$_p[\sigma]$-sentences that were developed by Niemistö in [Nie07] and that will be used later on as examples for the locality and non-locality properties of $\mathcal{ARB}$-inv-FO+MOD$_p[\sigma]$-sentences.

Example 3.4.1 (Niemistö (Proposition 6.22 in [Nie07]))**.** Let $\sigma = \{E\}$ be the signature consisting of a binary relation symbol E. This example presents an $<$-inv-FO+MOD$_2[\sigma]$-sentence $\varphi_{\textit{even cycles}}$ that is satisfied by exactly those finite σ-structures $\mathfrak{A}$ that are disjoint unions of directed cycles where the number of cycles of even length is even.

Clearly, a finite σ-structure $\mathfrak{A}$ is a disjoint union of directed cycles iff every element $a \in A$ has in-degree 1 and out-degree 1. That is, it is the graph of a permutation of A. This can easily be expressed by an FO$[\sigma]$-sentence $\varphi_{\textit{cycles}}$.

Each $\mathfrak{A}$ with $\mathfrak{A} \models \varphi_{cycles}$ can be identified with the permutation $\pi_{\mathfrak{A}}$ of A where, for every $a \in A$, $\pi_{\mathfrak{A}}(a) = b$ for the unique element $b \in A$ with $(a, b) \in E^{\mathfrak{A}}$. Note that the cycles of $\mathfrak{A}$ precisely correspond to the cycle decomposition of the permutation $\pi_{\mathfrak{A}}$.

Now let $n := |A|$, let ι be an arbitrary embedding of $\mathfrak{A}$ into $[n]$, and let $\pi^{\iota}_{\mathfrak{A}}$ be defined via $\pi^{\iota}_{\mathfrak{A}}(j) := \iota(\pi_{\mathfrak{A}}(\iota^{-1}(j)))$, for every $j \in [n]$. Clearly, $\pi^{\iota}_{\mathfrak{A}}$ is a permutation of $[n]$, and the cycle decomposition of $\pi^{\iota}_{\mathfrak{A}}$ is obtained from the cycle decomposition of $\pi_{\mathfrak{A}}$ by replacing every $a \in A$ with the number $\iota(a)$.

It is a well-known fact concerning permutation groups (see [Rot94]) that the cycle decomposition of $\pi^{\iota}_{\mathfrak{A}}$ has an even number of cycles of even length if, and only if, the number of *inversions*, i.e., pairs $(x, y) \in [n]^2$ with $x < y$ and $\pi^{\iota}_{\mathfrak{A}}(y) < \pi^{\iota}_{\mathfrak{A}}(x)$, is even. The latter can easily be expressed in FO+MOD$_2[\sigma, <]$ by the formula

$$\varphi_{inversions} \ := \ \exists^{0 \bmod 2} x \ \exists^{1 \bmod 2} y \ \big(\, x < y \ \wedge \ \pi(y) < \pi(x) \, \big),$$

where $\pi(y) < \pi(x)$ is an abbreviation for the formula $\exists x' \exists y' \big(E(x, x') \wedge E(y, y') \wedge \ y' < x'\big)$.

In summary, the formula $\varphi_{even\ cycles} := (\varphi_{cycles} \wedge \varphi_{inversions})$ is an $<$-inv-FO+MOD$_2[\sigma]$-sentence that is satisfied by exactly those finite σ-structures that are disjoint unions of cycles where the number of cycles of even length is even. $\square$

Example 3.4.2 (Niemistö (Proposition 6.20 in [Nie07])). Let $\sigma = \{E_1, E_2\}$ be the signature consisting of two binary relation symbols E_1 and E_2. Let $h \in \mathbb{N}$. A *torus* is a σ-structure $\mathfrak{T}$ with universe $[h] \times [w]$, for some $w \geqslant 2$, and relations

$$\begin{aligned} E_1^{\mathfrak{T}} \ &:= \ \{ \ \big((i, j), \ (i{+}1 \operatorname{rem} h, \ j)\big) \ : \ i \in [h], \ \ j \in [w] \ \} \\ E_2^{\mathfrak{T}} \ &:= \ \{ \ \big((i, j), \ (i, j{+}1)\big) \ : \ i \in [h], \ j \in [w{-}1] \ \} \ \cup \\ & \quad\ \ \{ \ \big((i, w{-}1) \, , \ (i{+}k \operatorname{rem} h, \ 0)\big) \ : \ i \in [h] \ \} \ , \ \ \text{for some } k \in [h]. \end{aligned}$$

The number w is the *width* of $\mathfrak{T}$. The set $[h] \times \{j\}$ is the *j-th column* of $\mathfrak{T}$, for each $j \in [w]$. The number h is the *height* of $\mathfrak{T}$ which will be fixed throughout this example. The number k is the *twist* of $\mathfrak{T}$ which we also denote by $twist(\mathfrak{T})$. See Figure 3.4.1 for an illustration of two tori with different twist.

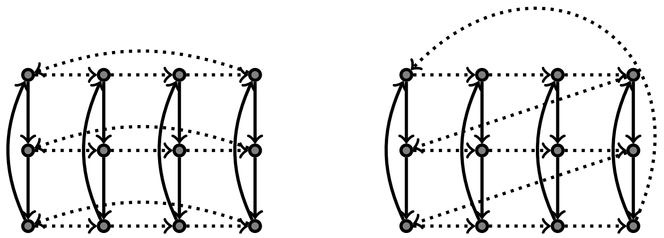
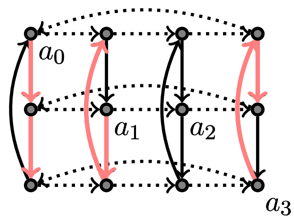


Figure 3.4.1: Tori of height 3 and width 4 with twist 0 (left) and twist 1 (right). The E_1- and E_2-edges are depicted by solid arcs and dotted arcs, respectively.

We consider disjoint unions of tori of height h. The *twist* of a disjoint union $\mathfrak{A}$ of tori $\mathfrak{T}_1, \ldots, \mathfrak{T}_\ell$ is defined as $twist(\mathfrak{A}) := (twist(\mathfrak{T}_1) + \cdots + twist(\mathfrak{T}_\ell)) \operatorname{rem} h$. This example presents an $<$-inv-FO+MOD$_h[\sigma]$-sentence φ_h which defines the class of disjoint unions of tori of height h and twist 0 on the class of finite σ-structures.

With the help of Hanf-locality of FO+MOD$_h$ (cf. [HLN99]), it is easy to show that plain FO+MOD$_h$ cannot distinguish between tori with twist 0 and tori with twist 1 if their width is sufficiently large, depending on the sentence. Here, we show that $<$-inv-FO+MOD$_h$ can distinguish between tori with twist 0 and tori with twist 1. To this end, we use the order to choose a distinguished element from each column of a torus.



A *list of representatives* for a torus $\mathfrak{T}$ of width w and height h is a tuple $R = (a_0, \dots, a_{w-1})$ of nodes of $\mathfrak{T}$ where, for each $j \in [w]$, a_j belongs to the j-th column of $\mathfrak{T}$. The *j-th turn path* with respect to R is the directed $E_1^{\mathfrak{T}}$-path from a_j to the $E_2^{\mathfrak{T}}$-successor of $a_{(j-1 \,\mathrm{rem}\, w)}$. The length of the j-turn path w.r.t R is called the *R-turn of the j-th column of* $\mathfrak{T}$, in symbols: $turn_j(\mathfrak{T}, R)$. Note that $turn_j(\mathfrak{T}, R) \in [h]$. The *$R$-turn of* $\mathfrak{T}$ is defined as $turn(\mathfrak{T}, R) := turn_0(\mathfrak{T}, R) + \cdots + turn_{w-1}(\mathfrak{T}, R)$. See the picture to the right for an illustration of a torus with a set of representatives R with R-turn 0; the turn paths are drawn thick and red.

Claim 3.4.3. $turn(\mathfrak{T}, R) \equiv twist(\mathfrak{T}) \,(mod\, h)$*, for every list* R *of representatives for* $\mathfrak{T}$*.*

Proof. Let $k := twist(\mathfrak{T})$. First note that if $R = (a_0, \dots, a_{w-1})$ is such that $a_j = (0, j)$, for every $j \in [w]$, then the following is true: $turn_j(\mathfrak{T}, R) = 0$ for every $j \in [1, w{-}1]$ (as a_j is the $E_2^{\mathfrak{T}}$-successor of a_{j-1}), and $turn_0(\mathfrak{T}, R) = k$ (as $a_0 = (0,0)$ and the $E_2^{\mathfrak{T}}$-successor of $(0, w{-}1)$ is $(k, 0)$). Thus, $turn(\mathfrak{T}, R) = k$.

Now consider arbitrary lists $R = (a_0, \dots, a_{w-1})$ of representatives. By induction on the number n_R of elements a_j such that $a_j \neq (0, j)$, we show that $turn(\mathfrak{T}, R) \equiv k \,(\mathrm{mod}\, h)$.

The induction base where $n_R = 0$ has been shown already. For the induction step, let j be such that $a_j = (p, j)$ for some $p \in [1, h{-}1]$. Let R' be obtained from R by replacing a_j with the element $(0, j)$. By induction we know that $turn(\mathfrak{T}, R') \equiv k \,(\mathrm{mod}\, h)$. By a straightforward case distinction, one can verify that the following is true: $turn_j(\mathfrak{T}, R') \equiv turn_j(\mathfrak{T}, R) + p \,(\mathrm{mod}\, h)$ and $turn_{j+1 \,\mathrm{rem}\, w}(\mathfrak{T}, R') \equiv turn_{j+1 \,\mathrm{rem}\, w}(\mathfrak{T}, R) - p \,(\mathrm{mod}\, h)$. Furthermore, $turn_{j'}(\mathfrak{T}, R') = turn_{j'}(\mathfrak{T}, R)$ for all $j' \in [w] \setminus \{\, j,\; j{+}1 \,\mathrm{rem}\, w \,\}$. Hence, $turn(\mathfrak{T}, R) \equiv turn(\mathfrak{T}, R') \,(\mathrm{mod}\, h)$. This completes the proof of the claim. $\square$

Now consider a disjoint union $\mathfrak{A}$ of tori $\mathfrak{T}_1, \dots, \mathfrak{T}_\ell$ of height h.
From the previous claim, we obtain

$$\begin{aligned} twist(\mathfrak{A}) &\equiv twist(\mathfrak{T}_1) + \cdots + twist(\mathfrak{T}_\ell) \,(\mathrm{mod}\, h) \\ &\equiv turn(\mathfrak{T}_1, R_1) + \cdots + turn(\mathfrak{T}_\ell, R_\ell) \,(\mathrm{mod}\, h), \end{aligned}$$

for all lists of representatives $R_1, \dots, R_\ell$ of $\mathfrak{T}_1, \dots, \mathfrak{T}_\ell$.

We proceed with the construction of an FO+MOD$_h[\sigma, <]$-sentence ψ_h which, when evaluated in a disjoint union of tori $\mathfrak{T}_1, \dots, \mathfrak{T}_\ell$ of height h and width $w_1, \dots, w_\ell$, computes $(turn(\mathfrak{T}_1, R_1) + \cdots + turn(\mathfrak{T}_\ell, R_\ell)) \,\mathrm{rem}\, h$, for lists of representatives $R_1, \dots, R_\ell$ of $\mathfrak{T}_1, \dots, \mathfrak{T}_\ell$ which depend on $<$. The sentence ψ_h is satisfied iff this sum modulo h is 0, i.e., iff

$twist(\mathfrak{A}) = 0$. The sentence uses the order $<$ only to choose a particular list of representatives $R = (a_0, \dots, a_{w-1})$ for each torus $\mathfrak{T}$ of the disjoint union $\mathfrak{A}$ by letting a_j be the smallest element of the j-th column. Hence, ψ_h is order-invariant on the class of all disjoint unions of tori of height h. To compute $twist(\mathfrak{A})$, the formula uses a formula $\psi_{\textit{on-turn-path}}(x)$ which is satisfied by exactly those elements b in the j-th column, for each $j \in [w]$, of a torus $\mathfrak{T}$ that are different from a_j and which lie on the j-turn path w.r.t R. Then, $twist(\mathfrak{T}) = turn(\mathfrak{T}, R)$ is exactly the number, modulo h, of all nodes b of $\mathfrak{T}$ for which $\mathfrak{T} \models \psi_{\textit{on-turn-path}}[b]$. Consequently, $twist(\mathfrak{A})$ is exactly the number, modulo h, of all nodes b of $\mathfrak{A}$ for which $\mathfrak{A} \models \psi_{\textit{on-turn-path}}[b]$. Thus, we can choose

$$\psi_h \ := \ \exists^{0 \bmod h} x \ \psi_{\textit{on-turn-path}}(x).$$

It remains to define the formula $\psi_{\textit{on-turn-path}}(x)$. Since the height h is fixed, it is easy to see that there are FO[σ]-formulae $\varphi_{\textit{col}}(x, y)$ and $\varphi_{\textit{between}}(x, y, z)$, and an FO[$\sigma, <$]-formula $\varphi_R(x)$, such that for each torus $\mathfrak{T}$ and all elements $a, b, c \in T$ the following holds:

- $\mathfrak{T} \models \varphi_{\textit{col}}[a, b]$ iff a and b belong to the same column of $\mathfrak{T}$.
- $\mathfrak{T} \models \varphi_{\textit{between}}[a, b, c]$ iff a, b, c belong to the same column, $b \neq a$, and b lies on the directed $E_1^{\mathfrak{T}}$-path of length at most $h{-}1$ from a to c.
- $\mathfrak{T} \models \varphi_R[a]$ iff a is the smallest element with respect to $<$ in its column.

Now, we can choose

$$\psi_{\textit{on-turn-path}}(x) \ := \quad \exists y \, \exists z \, \exists z' \ \big(\ \varphi_R(y) \ \wedge \ \varphi_R(z) \ \wedge \ E_2(z, z') \ \wedge \ \varphi_{\textit{between}}(y, x, z') \ \big).$$

Finally, we want to construct the sentence φ_h which is order-invariant on FIN_σ and which defines the class of disjoint unions of tori with twist 0. It is not difficult to see that a finite σ-structure $\mathfrak{A}$ is isomorphic to a disjoint union of tori iff it satisfies the following three properties:

- The relations $E_1^{\mathfrak{A}}$ and $E_2^{\mathfrak{A}}$ are graphs of permutations π_1 and π_2 of A (i.e. disjoint unions of cycles).
- The cycles of $E_1^{\mathfrak{A}}$, which we call *columns*, all have length h.
- The cycles of $E_2^{\mathfrak{A}}$ all have length at least 2.
- The π_2-image of each column $\mathfrak{C}$ is a column $\mathfrak{C}'$ and the restriction of π_2 to $\mathfrak{C}$ is an isomorphism of $\mathfrak{C}$ and $\mathfrak{C}'$.

Note that these properties can be expressed by an FO[σ]-sentence θ_h. For the last property, this can be done easily since the length h of the columns is fixed. We let $\varphi_h := \theta_h \wedge \psi_h$. Since ψ_h is order-invariant on disjoint unions of tori and θ_h defines the class of all disjoint unions of tori, φ_h is order-invariant on the class FIN_σ.

3.5 Locality of queries

In this section, we obtain our locality and non-locality results for general finite structures.

3.5.1 Gaifman locality

As we have discussed in the introduction of this chapter, the notion of *Gaifman locality* provides a standard tool for showing that particular queries are not definable in certain logics.

Definition 3.5.1 (Gaifman locality). Let $\mathfrak{C}$ be a class of finite σ-structures, $k \in \mathbb{N}_{\geqslant 1}$ and $f : \mathbb{N} \to \mathbb{N}$. A k-ary query q is *Gaifman $f(n)$-local on $\mathfrak{C}$* if there is an $n_0 \in \mathbb{N}$ such that for every $n \in \mathbb{N}$ with $n \geqslant n_0$ and every σ-structure $\mathfrak{A} \in \mathfrak{C}$ with $|A| = n$, the following is true for all k-tuples $\overline{a}, \overline{b} \in A^k$ with $\mathcal{N}^{\mathfrak{A}}_{f(n)}(\overline{a}) \cong \mathcal{N}^{\mathfrak{A}}_{f(n)}(\overline{b})$:

$$\overline{a} \in q(\mathfrak{A}) \iff \overline{b} \in q(\mathfrak{A}).$$

The query q is *Gaifman $f(n)$-local* if it is Gaifman $f(n)$-local on FIN_σ.

That is, in a σ-structure of cardinality n, a query that is Gaifman $f(n)$-local cannot distinguish between k-tuples of nodes whose $f(n)$-spheres are isomorphic. The function $f(n)$ is called the *locality radius* of the query. As a simple example of a query which is Gaifman $O(1)$-local, consider the unary query which maps an undirected graph to the set of all vertices which are the central vertex of a star with an even number of rays. Here, a *star* is a connected graph where all but the unique central vertex are rays, i.e. vertices of degree one. This query is Gaifman 2-local on the class of all undirected graphs. As an example of a query which is not Gaifman $f(n)$-local for any sublinear function f, consider the binary reachability query which maps a graph to all pairs of vertices such that there exists a path from the first vertex to the second vertex.

It is well-known that queries definable in FO or FO+MOD$_p$ (for any $p \geqslant 2$) are Gaifman local with a constant locality radius [HLN99]. The articles [GS00] and [AvMSS12] generalised this to order-invariant FO (for constant locality radius) and $\mathcal{ARB}$-invariant FO (for polylogarithmic locality radius) in the following sense: Let $k \in \mathbb{N}_{\geqslant 1}$, and let q be a k-ary query. If q is definable in $<$-inv-FO$[\sigma]$, then there is a $c \in \mathbb{N}$ such that q is Gaifman c-local. If q is definable in $\mathcal{ARB}$-inv-FO$[\sigma]$, then there is a $c \in \mathbb{N}$ such that q is Gaifman $(\log n)^c$-local. However, for every $d \in \mathbb{N}$ there is a unary query q_d that is definable in $\mathcal{ARB}$-inv-FO$[E]$ and that is not Gaifman $(\log n)^d$-local.

Somewhat surprisingly, using Example 3.4.2 one obtains that the Gaifman locality result cannot be generalised to order- or $\mathcal{ARB}$-invariant FO+MOD$_p$. In fact, $<$-inv-FO+MOD$_p$ can define queries that are not even Gaifman local with locality radius as big as $(\frac{n}{h}-2)$, for the smallest prime divisor h of p:

Proposition 3.5.2. *Let $h \in \mathbb{N}$ with $h \geqslant 2$, and let $\sigma = \{R, E_1, E_2\}$ be a signature consisting of a unary relation symbol R and two binary relation symbols E_1, E_2. There exists a unary query q that is not Gaifman $(\frac{n}{h}-2)$-local, but definable in $<$-inv-FO+MOD$_p[\sigma]$, for every multiple $p \geqslant 2$ of h.*

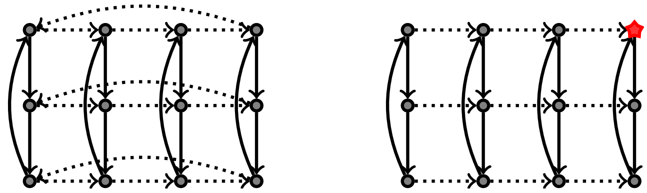


Figure 3.5.1: A torus $\mathfrak{T}$ (left) and the hose $\mathfrak{H}$ (right) obtained from $\mathfrak{T}$. The unique node in the relation $R^{\mathfrak{H}}$ of the hose is the rightmost node in the top row, depicted by a red star.

Proof. Recall the $\{E_1, E_2\}$-structures from Example 3.4.2 which we called tori. From a torus $\mathfrak{T}$ of height h and width w, we obtain a σ-structure $\mathfrak{H}$ by deleting all E_2-edges from the last to the first column and marking the least element of the last column with a unary relation. We call the structure $\mathfrak{H}$ a *hose*. More precisely, the universe of $\mathfrak{H}$ is $H := T = [h] \times [w]$ and the relations of $\mathfrak{H}$ are

$$\begin{aligned} E_1^{\mathfrak{H}} &:= E_1^{\mathfrak{T}}, \\ E_2^{\mathfrak{H}} &:= E_2^{\mathfrak{T}} \setminus \{ \big((i, w{-}1)\,,\,(i{+}\,twist(\mathfrak{T}) \operatorname{rem} h,\, 0)\big) \;:\; i \in [h] \}, \\ R^{\mathfrak{H}} &:= \{ (0, w{-}1) \}. \end{aligned}$$

From Example 3.4.2 we obtain an $<$-inv-FO+MOD$_h[E_1, E_2, <]$-sentence $\varphi_{h\text{-}torus}$ which defines the class of all tori $\mathfrak{T}$ of height h and $twist(\mathfrak{T}) = 0$ on FIN$_{\{E_1,E_2\}}$. We modify $\varphi_{h\text{-}torus}$ in such a way that we obtain an $<$-inv-FO+MOD$_h[\sigma, <]$-formula $\psi(x)$ which, when evaluated in a hose $\mathfrak{H}$ with x interpreted as the element $a := (0, 0)$ or $b := (1, 0)$, simulates $\varphi_{h\text{-}torus}$ evaluated on a torus of height h with twist 0 or twist 1, respectively. To this end, we let $\psi(x)$ state that each of the following is satisfied:

- There is a unique element y_0 satisfying $R(y_0)$,
- there are elements $y_1, \ldots, y_{h-1}$ such that $E_1(y_i, y_{i+1 \operatorname{rem} h})$ is true for all $i \in [h]$,
- there are elements $x_0, \ldots, x_{h-1}$ such that $x_0{=}x$ and $E_1(x_i, x_{i+1 \operatorname{rem} h})$ is true for all $i \in [h]$,
- the formula φ' is satisfied, where φ' is obtained from $\varphi_{h\text{-}torus}$ by replacing every atom of the form $E_2(u, v)$ by the formula $\big(E_2(u, v) \;\vee\; \bigvee_{0\leqslant i<h} (u{=}y_i \;\wedge\; v{=}x_i)\big)$.

Clearly, $\mathfrak{H} \models \psi[a]$ (since $\mathfrak{T} \models \varphi_{h\text{-torus}}$ if $twist(\mathfrak{T}) = 0$), and $\mathfrak{H} \not\models \psi[b]$ (since $\mathfrak{T} \not\models \varphi_{h\text{-torus}}$ if $twist(\mathfrak{T}) = 1$). Thus, $a \in q_\psi(\mathfrak{H})$ and $b \notin q_\psi(\mathfrak{H})$. Note that the $(w{-}2)$-spheres of a and b in the hose $\mathfrak{H}$ are isomorphic, i.e., $\mathcal{N}^{\mathfrak{H}}_{w-2}(a) \cong \mathcal{N}^{\mathfrak{H}}_{w-2}(b)$. The cardinality of $\mathfrak{H}$ is $n := hw$, and hence $w{-}2 = \frac{n}{h}{-}2$. Thus, the query defined by $\psi(x)$ is not Gaifman $(\frac{n}{h}{-}2)$-local.

By Example 3.4.2, $\varphi_{h\text{-}torus}$ is order-invariant on the class of all finite $\{E_1, E_2\}$-structures. Therefore, the formula $\psi(x)$ is order-invariant on FIN$_\sigma$. Furthermore, $\varphi_{h\text{-}torus}$ uses only modulo-counting quantifiers with modulus h and the construction of $\psi(x)$ adds no new modulo-counting quantifiers. As we observed in Section 3.3, if p is a multiple of h, each modulo-counting quantifier with modulus h can also expressed using quantifiers with modulus p. $\square$

3.5.2 Weak Gaifman locality

Weak Gaifman locality (cf., [Lib04]) is a relaxed notion of Gaifman locality where "$\overline{a} \in q(\mathfrak{A}) \iff \overline{b} \in q(\mathfrak{A})$" needs to be true only for those tuples $\overline{a}$ and $\overline{b}$ whose $f(n)$-spheres are disjoint.

Definition 3.5.3 (Weak Gaifman locality). Let $\mathfrak{C}$ be a class of finite σ-structures, $k \in \mathbb{N}_{\geqslant 1}$ and $f : \mathbb{N} \to \mathbb{N}$. A k-ary query q is *weakly Gaifman $f(n)$-local on* $\mathfrak{C}$ if there is an $n_0 \in \mathbb{N}$ such that for every $n \in \mathbb{N}$ with $n \geqslant n_0$ and every σ-structure $\mathfrak{A} \in \mathfrak{C}$ with $|A| = n$, the following is true for all k-tuples $\overline{a}, \overline{b} \in A^k$ with $\mathcal{N}^{\mathfrak{A}}_{f(n)}(\overline{a}) \cong \mathcal{N}^{\mathfrak{A}}_{f(n)}(\overline{b})$ and $N^{\mathfrak{A}}_{f(n)}(\overline{a}) \cap N^{\mathfrak{A}}_{f(n)}(\overline{b}) = \emptyset$: $\overline{a} \in q(\mathfrak{A}) \iff \overline{b} \in q(\mathfrak{A})$. The query q is *weakly Gaifman $f(n)$-local* if it is weakly Gaifman $f(n)$-local on FIN_σ.

Note that the example presented in the proof of Proposition 3.5.2 does not provide a counter-example to *weak* Gaifman locality, since the elements a and b considered in the proof of Proposition 3.5.2 are of distance 1, and thus their $f(n)$-neighbourhoods are not disjoint. However, using Example 3.4.1, one obtains a counter-example to *weak* Gaifman locality for $<$-inv-FO+MOD$_p$ for *even* numbers p; see Proposition 6.23 in [Nie07]. Here, we present a refinement of Niemistö's proof which provides a counter-example to weak Gaifman locality already for the restricted case of word structures.

Proposition 3.5.4. *Let $\Sigma := \{0,1\}$, and let $\sigma_\Sigma = \{E, P_0, P_1\}$ be the signature used for representing words over Σ. There exists a unary query q that is not weakly Gaifman $(\frac{n}{4}-1)$-local on Σ^+, but definable in $<$-inv-FO+MOD$_p[\sigma_\Sigma]$, for every even number $p \geqslant 2$.*

Proof. For every $\ell \in \mathbb{N}_{\geqslant 1}$, let $\mathfrak{A}_\ell$ and $\mathfrak{B}_\ell$ be $\{E\}$-structures whose universe consists of 2ℓ vertices, the edge relation of $\mathfrak{A}_\ell$ consists of two directed cycles of length ℓ, and the edge relation of $\mathfrak{B}_\ell$ consists of a single directed cycle of length 2ℓ. Furthermore, we choose w_ℓ to be the word $1^\ell\, 0^\ell\, 1^\ell\, 0^\ell$, and we let $a_\ell := \ell$ be the rightmost position of the first block of 1s, and $b_\ell := 3\ell$ the rightmost position of the second block of 1s.

From Example 3.4.1 we obtain an $<$-inv-FO+MOD$_2[E]$-sentence $\varphi_{\textit{even cycles}}$ that is satisfied by a finite $\{E\}$-structure $\mathfrak{A}$ iff $\mathfrak{A}$ is a disjoint union of directed cycles where the number of cycles of even length is even. Thus, for every $\ell \in \mathbb{N}_{\geqslant 1}$ we have: $\mathfrak{A}_\ell \models \varphi_{\textit{even cycles}}$ and $\mathfrak{B}_\ell \not\models \varphi_{\textit{even cycles}}$. We modify the formula $\varphi_{\textit{even cycles}}$ in such a way that we obtain an $<$-inv-FO+MOD$_2[\sigma_\Sigma]$-formula $\psi(x)$ which, when evaluated in the σ_Σ-structure w_ℓ representing the word w_ℓ with x interpreted as the position a_ℓ or b_ℓ simulates $\varphi_{\textit{even cycles}}$ evaluated on $\mathfrak{A}_\ell$ or $\mathfrak{B}_\ell$, respectively. To this end, we let $\psi(x)$ be a formula stating that each of the following is satisfied:

- There is a unique position $x' \neq x$ that carries the letter 1 such that the position directly to the right of x' carries the letter 0.

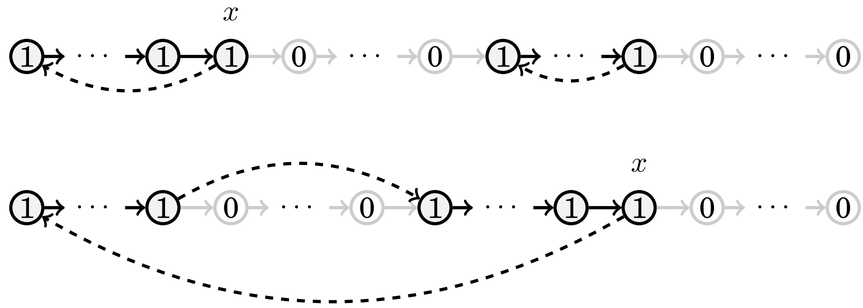


Figure 3.5.2: A word structure w_ℓ with $x = a_\ell$ (top) and $x = b_\ell$ (bottom). The formula $\psi(x)$ simulates $\varphi_{\textit{even cycles}}$ on the structure where the dashed edges are added and the light nodes and edges are removed.

- There is a unique position y of in-degree 0, and this position carries the letter 1. Furthermore, there is a unique position y' that carries the letter 1, such that the position directly to the left of y' carries the letter 0.
- The formula φ' is satisfied, where φ' is obtained from $\varphi_{\textit{even cycles}}$ by relativisation of all quantifiers to positions that carry the letter 1, and by replacing every atom of the form $E(u,v)$ by the formula $\big(E(u,v) \vee \big(u{=}x \wedge v{=}y\big) \vee \big(u{=}x' \wedge v{=}y'\big)\big)$.

Clearly, for every $\ell \in \mathbb{N}_{\geqslant 1}$ we have: $w_\ell \models \psi[a_\ell]$ (since $\mathfrak{A}_\ell \models \varphi_{\textit{even cycles}}$), and $w_\ell \not\models \psi[b_\ell]$ (since $\mathfrak{B}_\ell \not\models \varphi_{\textit{even cycles}}$). Thus, $a_\ell \in q_\psi(w_\ell)$ and $b_\ell \notin q_\psi(w_\ell)$. Note that the $(\ell{-}1)$-spheres of a_ℓ and b_ℓ in w_ℓ are disjoint and isomorphic. The cardinality of w_ℓ is $n := 4\ell$, and hence $\ell{-}1 = \frac{n}{4}{-}1$. Thus, the query defined by $\psi(x)$ is not weakly Gaifman $(\frac{n}{4}{-}1)$-local. Since $\varphi_{\textit{even cycles}}$ is order-invariant on all finite $\{E\}$-structures, the formula $\psi(x)$ is order-invariant on $\mathrm{FIN}_{\sigma_\Sigma}$. Note that $\psi(x)$ is definable in $<$-inv-FO+MOD$_2[\sigma_\Sigma]$, and hence also in $<$-inv-FO+MOD$_p[\sigma_\Sigma]$, for every multiple p of 2. □

In light of Proposition 3.5.4 it is somewhat surprising that for *odd* numbers p, unary queries definable in $<$-inv-FO+MOD$_p$ are weakly Gaifman local with constant locality radius — this is a result obtained by Niemistö (see Corollary 6.37 in [Nie07]). For *odd prime powers* p we can generalise this to k-ary queries definable in $\mathcal{ARB}$-inv-FO+MOD$_p$, when allowing polylogarithmic locality radius. Note that we cannot hope for a smaller locality radius, since [AvMSS12] provides, for every $d \in \mathbb{N}$, a unary query definable in $\mathcal{ARB}$-inv-FO$[E]$ that is not weakly Gaifman $(\log n)^d$-local. Precisely, we will show the following.

Theorem 3.5.5. *Let $\mathfrak{C}$ be a class of finite σ-structures. Let $k \in \mathbb{N}_{\geqslant 1}$, let q be a k-ary query, and let p be an odd prime power. If q is definable in $\mathcal{ARB}$-*inv-FO+MOD$_p[\sigma]$ *on $\mathfrak{C}$, then there is a $c \in \mathbb{N}$ such that q is weakly Gaifman $(\log n)^c$-local on $\mathfrak{C}$.*

The proof of this theorem will be given in the next subsection, as an easy consequence of Theorem 3.5.7 below. A generalisation of Theorem 3.5.5 from odd prime powers to arbitrary odd numbers p would lead to new separations concerning circuit complexity classes and can therefore be expected to be rather difficult (see Remark 3.5.14).

3.5.3 Shift locality

The following notion of *shift locality* generalises the notion of *alternating Gaifman locality* introduced by Niemistö in [Nie07]. In some sense discussed below, it unifies this notion and the notion of weak Gaifman locality.

Definition 3.5.6 (Shift locality)**.** Let $\mathfrak{C}$ be a class of finite σ-structures. Let $k, t \in \mathbb{N}_{\geqslant 1}$ with $t \geqslant 2$, and let $f : \mathbb{N} \to \mathbb{N}$. A kt-ary query q is *shift $f(n)$-local w.r.t. t on $\mathfrak{C}$* if there is an $n_0 \in \mathbb{N}$ such that for every $n \in \mathbb{N}$ with $n \geqslant n_0$ and every σ-structure $\mathfrak{A} \in \mathfrak{C}$ with $|A| = n$, the following is true for all k-tuples $\overline{a}_0, \ldots, \overline{a}_{t-1} \in A^k$ with $\mathcal{N}^{\mathfrak{A}}_{f(n)}(\overline{a}_i) \cong \mathcal{N}^{\mathfrak{A}}_{f(n)}(\overline{a}_j)$ and $N^{\mathfrak{A}}_{f(n)}(\overline{a}_i) \cap N^{\mathfrak{A}}_{f(n)}(\overline{a}_j) = \emptyset$ for all $i, j \in [t]$ with $i \neq j$: $\quad (\overline{a}_0, \overline{a}_1 \ldots, \overline{a}_{t-1}) \in q(\mathfrak{A}) \iff (\overline{a}_1, \ldots, \overline{a}_{t-1}, \overline{a}_0) \in q(\mathfrak{A})$.
Query q is *shift $f(n)$-local w.r.t. t* if it is shift $f(n)$-local w.r.t. t on FIN_σ.

The case of $k = 1$ and $t = 3$ for a constant function f yields Niemistö's notion of alternating Gaifman locality [Nie07]. The query q of Proposition 3.5.4 yields an example of an alternatingly Gaifman local query. For all even numbers $p \geqslant 2$, this query q is definable by an $<$-inv-FO+MOD$_p$-formula $\psi(x)$. The query defined by the formula $\varphi(x, y, z) := \psi(x) \wedge y = y \wedge z = z$ is an alternatingly Gaifman local query. This can be seen directly, but it is also a consequence of the result of [Nie07] that queries which are definable by formulae of $<$-inv-FO+MOD$_p$ for even numbers p are alternatingly Gaifman local. Several examples of non-shift locality will be given in Section 3.5.4. To understand the relation between shift locality and weak Gaifman locality, consider a k-ary query q and the $2k$-ary query $\tilde{q}$ with $\tilde{q}(\mathfrak{A}) := \{\overline{a}_1\overline{a}_2 : \overline{a}_1 \in q(\mathfrak{A}), \overline{a}_2 \in A^k\}$. Then q is weakly Gaifman $f(n)$-local iff $\tilde{q}$ is shift $f(n)$-local w.r.t. 2. The notion of shift locality helps to discuss both kinds of locality in a uniform way.

In a technical lemma (Lemma 6.36 in [Nie07]), Niemistö showed that for $k = 1$ and $p, t \in \mathbb{N}$ with $p, t \geqslant 2$ and p and t coprime, for every t-ary query q which is definable in $<$-inv-FO+MOD$_p$, there is a $c \in \mathbb{N}$ such that q is shift c-local w.r.t. t. Our next result deals with the general case of shift locality with a non-constant locality radius and more expressive formulae of $\mathcal{ARB}$-inv-FO+MOD$_p$.

Theorem 3.5.7. *Let $\mathfrak{C}$ be a class of finite σ-structures. Let $k, t \in \mathbb{N}_{\geqslant 1}$ with $t \geqslant 2$, let q be a kt-ary query, and let p be a prime power such that p and t are coprime. If q is definable in $\mathcal{ARB}$-inv-FO+MOD$_p[\sigma]$ on $\mathfrak{C}$, then there is a $c \in \mathbb{N}$ such that q is shift $(\log n)^c$-local w.r.t. t on $\mathfrak{C}$.*

Our proof of Theorem 3.5.7 relies on lower bounds achieved in circuit complexity. A generalisation of Theorem 3.5.7 from prime powers to arbitrary numbers $p \geqslant 2$ would lead to new separations of circuit complexity classes and can therefore be expected to be rather difficult (see Remark 3.5.14). Before giving the proof of Theorem 3.5.7, let us first point out that it immediately implies Theorem 3.5.5.

Proof of Theorem 3.5.5 (using Theorem 3.5.7). Let $\varphi(\overline{x}) \in \mathcal{ARB}$-inv-FO+MOD$_p[\sigma]$ with k free variables $\overline{x} = (x_1, \dots, x_k)$, defining a k-ary query q_φ on $\mathfrak{C}$. Let $\overline{y} = (y_1, \dots, y_k)$ be k variables different from the variables in $\overline{x}$. Then, $\psi(\overline{x}, \overline{y}) := \big(\varphi(\overline{x}) \wedge \bigwedge_{1 \leqslant i \leqslant k} y_i{=}y_i\big)$ is an $\mathcal{ARB}$-inv-FO+MOD$_p[\sigma]$-formula that defines a $2k$-ary query q_ψ. By Theorem 3.5.7, there exists a $c \in \mathbb{N}$ such that q_ψ is shift $(\log n)^c$-local w.r.t. $t := 2$ on $\mathfrak{C}$. It is straightforward to see that the shift $(\log n)^c$-locality of q_ψ w.r.t. $t = 2$ implies that the query q_φ is weakly Gaifman $(\log n)^c$-local. □

The remainder of this subsection is devoted to the proof of Theorem 3.5.7. We follow the overall method of [AvMSS12] for the case of disjoint neighbourhoods (see [Sch13] for an overview) and make use of the connection between $\mathcal{ARB}$-inv-FO+MOD$_p$ and MOD$_p$-circuits [BIS90], along with a circuit lower bound by Smolensky [Smo87].

We assume that the reader is familiar with basic notions and results in circuit complexity (cf., e.g., the textbook [AB09]). A MOD$_p$-gate outputs the value 1 iff the number of ones at its inputs is congruent 0 modulo p. We consider Boolean MOD$_p$-*circuits* consisting of AND-, OR-, and MOD$_p$-gates of unbounded fan-in, input gates, negated input gates, and constant gates **0** and **1**. More precisely, a MOD$_p$-*circuit with* m *inputs* is a directed acyclic graph whose vertices are called *gates* and which satisfies the following properties:

1. Each gate without ingoing edges (called an *input*) is labelled with either **0**, **1**, w_ν, or $\neg w_\nu$ for $\nu \in [1, m]$.
2. Each gate with ingoing edges is labelled either by AND, by OR, or by MOD$_p$.
3. There is one distinguished gate without outgoing edges called the *output gate*.

Such a circuit C takes an input word $w \in \{0,1\}^m$ and computes a value $C(w)$ by recursively computing the value of each gate. The input gates are assigned values according to the input word w and the value of each inner gate is computed according to its label from the values at its predecessor gates. The output $C(w)$ is then the value computed at the output gate of C. We say that C *accepts* w if $C(w) = 1$. Accordingly, C *rejects* w if $C(w) = 0$. The *size* of a circuit is the number of gates it contains, and the *depth* is the length of the longest path from any of the input gates to the output gate.

Our proof of Theorem 3.5.7 relies on Smolensky's following circuit lower bound.

Theorem 3.5.8 (Smolensky [Smo87] (see also [Str94]))**.** *Let p be a prime power and let r be a number which has a prime factor which is different from the prime factor of p. There exist numbers $\varepsilon, \ell > 0$ such that for every $d \in \mathbb{N}_{\geqslant 1}$ there is an $m_d \in \mathbb{N}_{\geqslant 1}$ such that for every $m \in \mathbb{N}$ with $m \geqslant m_d$ the following is true: No* MOD$_p$*-circuit of depth d and size at most $2^{\varepsilon \sqrt[\ell d]{m}}$ accepts exactly those binary words $w \in \{0,1\}^m$ that contain a number of ones congruent 0 modulo r.*

In the literature, Smolensky's theorem is usually stated only for primes p. Note, however, that (for each fixed $k \in \mathbb{N}_{\geqslant 1}$) MOD$_{p^k}$-gates can easily be simulated by MOD$_p$-circuits of constant depth and polynomial size (cf., [Str94]), and hence Smolensky's theorem also holds for prime powers p, as stated in Theorem 3.5.8. It is still open whether an analogous result also holds for numbers p composed of more than one prime factor (see Chapter VIII of [Str94] and Chapter 14.4 of [AB09] for discussions on this).

To establish the connection between MOD_p-circuits and $\mathcal{ARB}$-inv-FO+$\mathrm{MOD}_p[\sigma]$, we need to represent σ-structures $\mathfrak{A}$ and K-tuples $\overline{a} \in A^K$ (for $K \in \mathbb{N}$) by binary words. This is done in a straightforward way: Let $\sigma = \{R_1, \ldots, R_{|\sigma|}\}$ and let $r_i := \mathrm{ar}(R_i)$ for each $i \leqslant |\sigma|$. Consider a finite σ-structure $\mathfrak{A}$ with $|A| = n$. Let ι be an embedding of $\mathfrak{A}$. For each $R_i \in \sigma$ we let $\mathrm{Rep}^\iota(R_i^{\mathfrak{A}})$ be the binary word of length n^{r_i} whose j-th bit is 1 iff the j-th smallest element in A^{r_i} w.r.t. the lexicographic order associated with $<^\iota$ belongs to the relation $R_i^{\mathfrak{A}}$. Similarly, for each component a_i of a K-tuple $\overline{a} = (a_1, \ldots, a_K) \in A^K$ we let $\mathrm{Rep}^\iota(a_i)$ be the binary word of length n whose j-th bit is 1 iff a_i is the j-th smallest element of A w.r.t. $<^\iota$. Finally, we let

$$\mathrm{Rep}^\iota(\mathfrak{A}, \overline{a}) \ := \ \mathrm{Rep}^\iota(R_1^{\mathfrak{A}}) \ \cdots \ \mathrm{Rep}^\iota(R_{|\sigma|}^{\mathfrak{A}}) \, \mathrm{Rep}^\iota(a_1) \ \cdots \ \mathrm{Rep}^\iota(a_K)$$

be the *binary representation of* $(\mathfrak{A}, \overline{a})$ *w.r.t.* ι. Note that, independently of ι, the length of the binary word $\mathrm{Rep}^\iota(\mathfrak{A}, \overline{a})$ is $\lambda_K^\sigma(n) := \sum_{i=1}^{|\sigma|} n^{r_i} + Kn$.

The connection between FO+$\mathrm{MOD}_p[\sigma, \mathcal{ARB}]$ and MOD_p-circuits is obtained by the following result.

Theorem 3.5.9 (implicit in [BIS90] (see also [Str94])). *Let σ be a finite signature, let $K \in \mathbb{N}$, and let $p \in \mathbb{N}$ with $p \geqslant 2$. For every* FO+$\mathrm{MOD}_p[\sigma, \mathcal{ARB}]$*-formula $\varphi(\overline{x})$ with K free variables there exist numbers $d, s \in \mathbb{N}$ such that for every $n \in \mathbb{N}_{\geqslant 1}$ there is a* MOD_p*-circuit C_n with $\lambda_K^\sigma(n)$ inputs, depth d, and size n^s such that the following is true for all σ-structures $\mathfrak{A}$ with $|A| = n$, all $\overline{a} \in A^K$, and all embeddings ι of $\mathfrak{A}$:*

$$C_n \textit{ accepts } \mathrm{Rep}^\iota(\mathfrak{A}, \overline{a}) \iff \mathfrak{A}^\iota \models \varphi[\overline{a}].$$

The proof of Theorem 3.5.7 can be outlined informally as follows. Assume that, for some prime power p and a number t which is coprime with p, there is an $\mathcal{ARB}$-inv-FO+$\mathrm{MOD}_p[\sigma]$-definable kt-ary query q which is not shift $(\log n)^c$-local w.r.t t for each c. We wish to obtain a contradiction to Smolensky's theorem (Theorem 3.5.8). From the defining $\mathcal{ARB}$-invariant sentence for q, we obtain a MOD_p-circuit-family $(C_n)_{n \in \mathbb{N}}$ which accepts exactly the binary words which are representations of structures $(\mathfrak{A}, \overline{a})$ such that $\overline{a} \in q(\mathfrak{A})$ (cf. Theorem 3.5.9). Since q is not shift $(\log n)^c$-local w.r.t t, for each c, we obtain an infinite family of counter examples to the shift $(\log n)^c$-locality of q. That is, for each n_0 there is a structure $\mathfrak{A}$ on $n \geqslant n_0$ elements which contains tuples $\overline{a}_0, \ldots, \overline{a}_{t-1}$ of k elements each such that the tuples have pairwise isomorphic and disjoint $(\log n)^c$-spheres, but $(\overline{a}_0, \ldots, \overline{a}_{t-1}) \in q(\mathfrak{A})$ and $(\overline{a}_1, \ldots, \overline{a}_{t-1}, \overline{a}_0) \notin q(\mathfrak{A})$, i.e. C_n accepts all representations of $(\mathfrak{A}, \overline{a}_0, \ldots, \overline{a}_{t-1})$ and rejects all representations of $(\mathfrak{A}, \overline{a}_1, \ldots, \overline{a}_{t-1}, \overline{a}_0)$. We transform each C_n into a circuit $\tilde{C}_m$ (cf. Lemma 3.5.10 below) of exactly the same size which accepts all binary words where the number of ones is 0 modulo t and rejects all binary words where this number is 1 modulo t; furthermore, it does not distinguish between words with the same number of ones modulo t. To this end, on input of a word w, the circuit $\tilde{C}_m$ simulates the circuit C_n on the representation of a structure which is obtained from $\mathfrak{A}$ as follows. For each position i of w which is labelled by 1, the relations between elements at distance i and $i+1$ from the tuple $(\overline{a}_0, \ldots, \overline{a}_{t-1})$ are changed in such a way that the resulting structure looks like $\mathfrak{A}$ where the tuple $(\overline{a}_0, \ldots, \overline{a}_{t-1})$ has been shifted. See Figure 3.5.3 for an example where the structure $\mathfrak{A}$ is a graph. This will be done in a way which ensures that $\tilde{C}_m$ accepts w if the number of ones is 0 modulo t — in this case, the simulated structure looks like $(\mathfrak{A}, \overline{a}_0, \ldots, \overline{a}_{t-1})$ and hence its representation is accepted by C_n — and that it rejects w if the number of ones is 1 modulo t — in this case, the simulated structure looks like $(\mathfrak{A}, \overline{a}_1, \ldots, \overline{a}_{t-1}, \overline{a}_0)$ and hence it is rejected by C_n.

This construction is not yet sufficient for a contradiction to Smolenky's theorem. For this, we need to show (cf. Lemma 3.5.11 below) that the circuit $\tilde{C}_m$ can be transformed into a circuit $\hat{C}_m$ of roughly the same size and depth which accepts *exactly* the binary words with 0 ones modulo r, for some factor $r \geqslant 2$ of t. To this end, we show that there exists a factor $r \geqslant 2$ of t such that, basically, it is possible to determine if the number of ones in a binary word w is 0 modulo r by computing, for each $j \in [t]$, whether $\tilde{C}_m$ accepts w after replacing the first j zeros of w by ones. Having achieved all this, if we fix c and n_0 appropriately in terms of Smolenky's theorem, we obtain the desired contradiction.

We proceed with the formal proof of Theorem 3.5.7. To simplify notation, we define

$$\begin{aligned} \bar{a}^{(0)} &:= (\overline{a}_0, \overline{a}_1, \ldots, \overline{a}_{t-1}), \\ \bar{a}^{(i)} &:= (\overline{a}_i, \overline{a}_{i+1}, \ldots, \overline{a}_{t-1}, \overline{a}_0, \overline{a}_1, \ldots, \overline{a}_{i-1}), \end{aligned}$$

for all $i \in [t]$ with $i \leqslant 1$.

Lemma 3.5.10. *Let $m, k, t \in \mathbb{N}_{\geqslant 1}$ with $t \geqslant 2$. Let $\mathfrak{A}$ be a finite σ-structure with $n := |A|$. For each $i \in [t]$, let $\overline{a}_i \in A^k$ such that for all $i, j \in [t]$ with $i \neq j$ we have $(\mathcal{N}_m^{\mathfrak{A}}(\overline{a}_i) \cong \mathcal{N}_m^{\mathfrak{A}}(\overline{a}_j)$ and $N_m^{\mathfrak{A}}(\overline{a}_i) \cap N_m^{\mathfrak{A}}(\overline{a}_j) = \emptyset$. Let $p \in \mathbb{N}$ with $p \geqslant 2$. Let C be a MOD_p-circuit with $\lambda_{kt}^{\sigma}(n)$ inputs such that:*

1. *C accepts $\mathrm{Rep}^{\iota_1}(\mathfrak{A}, \bar{a}^{(i)})$ iff it accepts $\mathrm{Rep}^{\iota_2}(\mathfrak{A}, \bar{a}^{(i)})$, for all embeddings ι_1 and ι_2 of $\mathfrak{A}$ and for every $i \in [t]$, and*
2. *C accepts $\mathrm{Rep}^{\iota}(\mathfrak{A}, \bar{a}^{(0)})$ and rejects $\mathrm{Rep}^{\iota}(\mathfrak{A}, \bar{a}^{(1)})$, for every embedding ι of $\mathfrak{A}$.*

There exists a MOD_p-circuit $\tilde{C}$ with m inputs, such that:

(a) *$\tilde{C}$ has the same depth and size as C,*

(b) *for all $w, w' \in \{0,1\}^m$ with $|w|_1 \equiv |w'|_1 \,(\mathrm{mod}\, t)$, $\tilde{C}$ accepts w iff it accepts w', and*

(c) *$\tilde{C}$ accepts all $w \in \{0,1\}^m$ with $|w|_1 \equiv 0 \,(\mathrm{mod}\, t)$ and rejects all $w \in \{0,1\}^m$ with $|w|_1 \equiv 1 \,(\mathrm{mod}\, t)$.*

Proof. Let $I \subset [t]$ be the set containing $i \in [t]$ iff C accepts $\mathrm{Rep}^{\iota_1}(\mathfrak{A}, \bar{a}^{(i)})$ for some (i.e., due to property (1) of C, *every*) embedding ι_1 of $\mathfrak{A}$. By property (2) of C, we know that $0 \in I$ and $1 \notin I$.

For the remainder of this proof, fix an embedding ι of $\mathfrak{A}$. Note that ι is also an embedding of any other σ-structure that has the same universe as $\mathfrak{A}$. For every $w \in \{0,1\}^m$, we will

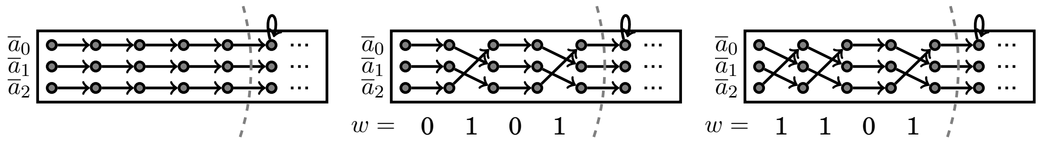


Figure 3.5.3: Illustration of a structure $\mathfrak{A}$ (left) and structures $\mathfrak{A}_w$ for two different binary words w.

define a σ-structure $\mathfrak{A}_w$ with the same universe as $\mathfrak{A}$, which has the following property for every $i \in [t]$:

$$\text{If } |w|_1 \equiv i \,(\text{mod}\, t), \text{ then } (\mathfrak{A}_w, \bar{a}^{(0)}) \cong (\mathfrak{A}, \bar{a}^{(i)}). \tag{3.1}$$

Note that if $(\mathfrak{A}_w, \bar{a}^{(0)}) \cong (\mathfrak{A}, \bar{a}^{(i)})$, then there is an embedding ι_1 such that $\text{Rep}^{\iota_1}(\mathfrak{A}, \bar{a}^{(i)}) = \text{Rep}^{\iota}(\mathfrak{A}_w, \bar{a}^{(0)})$. Thus, due to property (1), C accepts $\text{Rep}^{\iota}(\mathfrak{A}_w, \bar{a}^{(0)})$ iff it accepts $\text{Rep}^{\iota}(\mathfrak{A}, \bar{a}^{(i)})$.

The circuit $\tilde{C}$ will be constructed so that on input $w \in \{0,1\}^m$ it does the same as circuit C does on input $\text{Rep}^{\iota}(\mathfrak{A}_w, \bar{a}^{(0)})$. Thus, the following will be true for every $w \in \{0,1\}^m$ and the particular number $i \in [t]$ such that $|w|_1 \equiv i \,(\text{mod}\, t)$:

$$\tilde{C} \text{ accepts } w \iff C \text{ accepts } \text{Rep}^{\iota}(\mathfrak{A}_w, \bar{a}^{(0)}) \iff C \text{ accepts } \text{Rep}^{\iota}(\mathfrak{A}, \bar{a}^{(i)}) \iff i \in I.$$

This immediately implies that $\tilde{C}$ satisfies property (b); and since $0 \in I$ and $1 \notin I$, the circuit $\tilde{C}$ also satisfies property (c).

Definition of $\mathfrak{A}_w$*:* For each $j \in [t]$, we partition $N_m^{\mathfrak{A}}(\overline{a}_j)$ into *shells* $S_\nu(\overline{a}_j) := \{x \in A : \text{dist}^{\mathfrak{A}}(x, \overline{a}_j) = \nu\}$, for all $\nu \in \{0, \dots, m\}$. We write S_ν for the set $S_\nu(\overline{a}_0) \cup \cdots \cup S_\nu(\overline{a}_{t-1})$. For each $j \in [t]$ let π_j be an isomorphism from $\mathcal{N}_m^{\mathfrak{A}}(\overline{a}_j)$ to $\mathcal{N}_m^{\mathfrak{A}}(\overline{a}_{(j+1 \,\text{rem}\, t)})$. Note that $\pi_j(S_\nu(\overline{a}_j)) = S_\nu(\overline{a}_{(j+1 \,\text{rem}\, t)})$ for each $j \in [t]$ and each $\nu \leqslant m$.

For a binary word $w = w_1 \cdots w_m \in \{0,1\}^m$ the structure $\mathfrak{A}_w$ has the same universe as $\mathfrak{A}$. For each $R \in \sigma$ of arity r, the relation $R^{\mathfrak{A}_w}$ is obtained from $R^{\mathfrak{A}}$ as follows: We start with $R^{\mathfrak{A}_w} := \emptyset$, and then for each tuple $\overline{c} \in R^{\mathfrak{A}}$ we insert the tuple $\overline{c}_w$ into $R^{\mathfrak{A}_w}$, where $\overline{c}_w$ is defined as follows:

(*i*) If $\overline{c} \notin (S_{\nu-1} \cup S_\nu)^r$ for any $\nu \leqslant m$, or $\overline{c} \in S_\nu^r$ for some $\nu \leqslant m$, then $\overline{c}_w := \overline{c}$.

(*ii*) Otherwise, if $\overline{c} \in (S_{\nu-1} \cup S_\nu)^r$ for some $\nu \leqslant m$, then note that (since $\overline{c} \in R^{\mathfrak{A}}$), there is a unique $j \in [t]$ such that $\overline{c} \in (S_{\nu-1}(\overline{a}_j) \cup S_\nu(\overline{a}_j))^r$ (since $N_m^{\mathfrak{A}}(\overline{a}_j) \cap N_m^{\mathfrak{A}}(\overline{a}_{j'}) = \emptyset$, for all $j, j' \in [t]$ with $j \neq j'$). To keep the notation simple, assume that $\overline{c} = (\overline{c}_{\nu-1}, \overline{c}_\nu)$, where all elements of $\overline{c}_{\nu-1}$ belong to $S_{\nu-1}(\overline{a}_j)$ and all elements of $\overline{c}_\nu$ belong to $S_\nu(\overline{a}_j)$. We define $\overline{c}_w$ depending on the ν-th bit w_ν of w: If $w_\nu = 0$, then $\overline{c}_w := \overline{c}$. If $w_\nu = 1$, then $\overline{c}_w := (\overline{c}_{\nu-1}, \pi_j(\overline{c}_\nu))$.

Note that for every $\nu \in [1, m]$ with $w_\nu = 1$, this construction enforces that the role that was formerly played by shell $S_\nu(\overline{a}_j)$ is afterwards played by shell $S_\nu(\overline{a}_{(j+1 \,\text{rem}\, t)})$; see Figure 3.5.3 for an illustration. This shows that $\mathfrak{A}_w$ satisfies the property (3.1). A more precise argument will be given at the end of the proof.

Construction of $\tilde{C}$: On input of $w \in \{0,1\}^m$ the circuit $\tilde{C}$ will simulate circuit C on input $\mathrm{Rep}^\iota(\mathfrak{A}_w, \bar{a}^{(0)})$. We construct $\tilde{C}$ in a way which mirrors the construction of $\mathfrak{A}_w$. To this end, all inputs of C corresponding, in $\mathrm{Rep}^\iota(\mathfrak{A}, \bar{a}^{(0)})$, to tuples in relations of $\mathfrak{A}$ that are unchanged by the construction of $\mathfrak{A}_w$ (i.e. tuples to which case (i) of the definition of $\mathfrak{A}_w$ applies) are fixed to constant values. Inputs of C corresponding to tuples to which case (ii) of the definition of $\mathfrak{A}_w$ applies (i.e. tuples $\bar{c} \in (S_{\nu-1} \cup S_\nu)^r$ for some $\nu \leqslant m$) are changed according to the ν-th bit in the input w of $\tilde{C}$ in a way that mirrors case (ii).

More precisely, for each relation symbol $R \in \sigma$ of arity r and each tuple $\bar{c} \in R^{\mathfrak{A}}$, we proceed as follows. Let g and $\neg g$ be the non-negated and the negated input gate of C that correspond to the bit b in $\mathrm{Rep}^\iota(\mathfrak{A}, \bar{a}^{(0)})$ which represents the information whether $\bar{c}$ belongs to $R^{\mathfrak{A}}$.

(*i*) If $\bar{c} \notin (S_{\nu-1} \cup S_\nu)^r$ for all $\nu \leqslant m$, or $\bar{c} \in S_\nu^r$ for some $\nu \leqslant m$, the gate g is replaced by the constant gate **1**, and the negated input gate $\neg g$ is replaced by the constant gate **0**.

(*ii*) Otherwise, if $\bar{c} \in (S_{\nu-1}(\bar{a}_j) \cup S_\nu(\bar{a}_j))^r$ for some $\nu \leqslant m$ and $j \in [t]$, assume, for simplicity, that $\bar{c} := (\bar{c}_{\nu-1}, \bar{c}_\nu)$, where all elements of $\bar{c}_{\nu-1}$ belong to $S_{\nu-1}(\bar{a}_j)$ and all elements of $\bar{c}_\nu$ belong to $S_\nu(\bar{a}_j)$. Let g' and $\neg g'$ be the non-negated and the negated input gate of C that correspond to the bit b' in $\mathrm{Rep}^\iota(\mathfrak{A}, \bar{a}^{(0)})$ representing the information whether $(\bar{c}_{\nu-1}, \pi_j(\bar{c}_\nu))$ belongs to $R^{\mathfrak{A}}$.

- We replace gate g by the new input gate $\neg w_\nu$, and gate g' by the input gate w_ν.
- Accordingly, the negated input gates $\neg g$ and $\neg g'$ are replaced by the input gates w_ν and $\neg w_\nu$.

Afterwards, we consider all non-negated input gates g of C that have not yet been replaced, and we let b be the bit of $\mathrm{Rep}^\iota(\mathfrak{A}, \bar{a}^{(0)})$ that is inserted at input gate g. We replace g by the constant gate **1** if $b = 1$, and by the constant gate **0** if $b = 0$. Accordingly, the negated input gate $\neg g$ is replaced by **0** if $b = 1$, and by **1** if $b = 0$.

It is easy to see that the resulting circuit $\tilde{C}$ does the same on input $w \in \{0,1\}^m$ as circuit C does on input $\mathrm{Rep}^\iota(\mathfrak{A}_w, \bar{a}^{(0)})$. Furthermore, $\tilde{C}$ obviously has the same depth as C, and the size of $\tilde{C}$ is smaller than or equal to the size of C.

To finish the proof of Lemma 3.5.10, it remains to show that $\mathfrak{A}_w$ satisfies the property (3.1). For each $\nu \leqslant m$, we let $i_\nu := |w_1 \dots w_\nu|_1 \operatorname{rem} t$. In particular, $i_0 = 0$. For each map f and each subset X of its domain, we let $f \restriction X$ denote the restriction of f to X. By id_X we denote the identity map on a set X. For each $\nu \leqslant m$, we let $\mathrm{id}_\nu := \mathrm{id}_{S_\nu^{\mathfrak{A}_w}(\bar{a}^{(0)})}$. Here we have extended our notation for shells which was defined above with respect to $\mathfrak{A}$ to $\mathfrak{A}_w$. We write $g \circ f$ for the composition of maps $f : X \to Y$ and $g : Y \to Z$. Recall the maps $\pi_1, \dots, \pi_t$ from the definition of $\mathfrak{A}_w$ and observe that, since these maps have disjoint domains and images, their union π is a well-defined map, and that this map is an isomorphism from $\mathcal{N}_m^{\mathfrak{A}}(\bar{a}^{(i)})$ to $\mathcal{N}_m^{\mathfrak{A}}(\bar{a}^{(i+1 \operatorname{rem} t)})$, for each $i \in [t]$. For each $\nu \leqslant m$, we let $\gamma_\nu := \pi \restriction N_\nu^{\mathfrak{A}}(\bar{a}^{(0)})$ (which is the same as $\pi \restriction N_\nu^{\mathfrak{A}}(\bar{a}^{(i)})$, for each $i \in [t]$).

By induction on $\nu \leqslant m$, we prove that $N_\nu^{\mathfrak{A}_w}(\bar{a}^{(0)}) = N_\nu^{\mathfrak{A}}(\bar{a}^{(i_\nu)})$ and that there exists a map ρ_ν such that

$$\rho_\nu : \mathcal{N}_\nu^{\mathfrak{A}_w}(\bar{a}^{(0)}) \;\cong\; \mathcal{N}_\nu^{\mathfrak{A}}(\bar{a}^{(i_\nu)}). \tag{3.2}$$

For $\nu = 0$, we let $\rho_0 := \mathrm{id}_0$. If $1 \leqslant \nu \leqslant m$, we let

$$\rho_\nu := \begin{cases} \rho_{\nu-1} \cup \mathrm{id}_\nu & \text{if } w_\nu = 0, \\ (\gamma_{\nu-1} \circ \rho_{\nu-1}) \cup \mathrm{id}_\nu & \text{if } w_\nu = 1. \end{cases}$$

Before we show that ρ_ν satisfies (3.2), for each $\nu \leqslant m$, we show how to obtain the isomorphism ρ from $(\mathfrak{A}_w, \bar{a}^{(0)})$ to $(\mathfrak{A}, \bar{a}^{(i_m)})$ required by (3.1) if ρ_ν satisfies (3.2). We let $\rho := \rho_m \cup \mathrm{id}_{A \setminus N_m^{\mathfrak{A}_w}(\bar{a}^{(0)})}$, which is a bijection since ρ_m is a bijection from $N_m^{\mathfrak{A}_w}(\bar{a}^{(0)})$ to $N_m^{\mathfrak{A}}(\bar{a}^{(i_m)}) = N_m^{\mathfrak{A}_w}(\bar{a}^{(0)})$. According to (3.2), the restriction of ρ to $N_m^{\mathfrak{A}_w}(\bar{a}^{(0)})$ is an isomorphism. The restriction of ρ to $A \setminus N_m^{\mathfrak{A}_w}(\bar{a}^{(0)})$ is an isomorphism since the definition of $\mathfrak{A}_w$ (according to its case (i)) includes a tuple of elements which all have distance greater than m to $\bar{a}^{(0)}$ in a relation of $\mathfrak{A}_w$ iff it is included in the corresponding relation of $\mathfrak{A}$. To verify that ρ is an isomorphism from $(\mathfrak{A}_w, \bar{a}^{(0)})$ to $(\mathfrak{A}, \bar{a}^{(i_m)})$, it suffices to show that it preserves the relations between elements from $S_m^{\mathfrak{A}_w}$ and $S_{m+1}^{\mathfrak{A}_w}$. This follows since, restricted to $S_m^{\mathfrak{A}_w} \cup S_{m+1}^{\mathfrak{A}_w}$, the map ρ is the identity.

Now we prove the construction of ρ_ν correct. For $\nu = 0$, we have $i_0 = 0$ and from the construction of $\mathfrak{A}_w$, we see that $\mathcal{N}_0^{\mathfrak{A}_w}(\bar{a}^{(0)}) = \mathcal{N}_0^{\mathfrak{A}}(\bar{a}^{(0)})$. Hence, $\mathrm{id}_0 : \mathcal{N}_\nu^{\mathfrak{A}_w}(\bar{a}^{(0)}) \;\cong\; \mathcal{N}_\nu^{\mathfrak{A}}(\bar{a}^{(i_\nu)})$. Suppose now that $\nu \geqslant 1$. First, we consider the case that $w_\nu = 0$ and hence $i_\nu = i_{\nu-1}$. By induction, $N_{\nu-1}^{\mathfrak{A}_w}(\bar{a}^{(0)}) = N_{\nu-1}^{\mathfrak{A}}(\bar{a}^{(i_\nu)})$ and $\rho_{\nu-1} : \mathcal{N}_{\nu-1}^{\mathfrak{A}_w}(\bar{a}^{(0)}) \;\cong\; \mathcal{N}_{\nu-1}^{\mathfrak{A}}(\bar{a}^{(i_\nu)})$. If $w_\nu = 0$, case (ii) of the definition of $\mathfrak{A}_w$ includes a tuple $\bar{c}$ with elements from $S_{\nu-1}^{\mathfrak{A}_w} \cup S_\nu^{\mathfrak{A}_w}$ in a relation of $\mathfrak{A}_w$ iff it is included in the corresponding relation in $\mathfrak{A}$. Hence, we obtain $N_\nu^{\mathfrak{A}_w}(\bar{a}^{(0)}) = N_\nu^{\mathfrak{A}}(\bar{a}^{(i_\nu)})$. By definition of ρ_ν for $w_\nu = 0$, we have $\rho_\nu \restriction N_{\nu-1}^{\mathfrak{A}_w}(\bar{a}^{(0)}) = \rho_{\nu-1}$ and hence $\rho_\nu \restriction S_{\nu-1}^{\mathfrak{A}_w}(\bar{a}^{(0)}) = \mathrm{id}_{\nu-1}$. Since also $\rho_\nu \restriction S_\nu^{\mathfrak{A}_w}(\bar{a}^{(0)}) = \mathrm{id}_\nu$, we obtain $\rho_\nu : \mathcal{N}_\nu^{\mathfrak{A}_w}(\bar{a}^{(0)}) \;\cong\; \mathcal{N}_\nu^{\mathfrak{A}}(\bar{a}^{(i_\nu)})$.

Now consider the case that $w_\nu = 1$ and hence $i_\nu \equiv i_{\nu-1} + 1 \pmod{t}$. Since, by induction, (3.2) holds for $\nu - 1$, we can actually apply $\gamma_{\nu-1}$ after $\rho_{\nu-1}$, and hence ρ_ν is a well-defined bijection of $N_\nu^{\mathfrak{A}_w}(\bar{a}^{(0)})$ and $N_\nu^{\mathfrak{A}}(\bar{a}^{(i_\nu)})$. Furthermore, we have $\rho_{\nu-1}(\bar{a}_j) = \bar{a}_{(j+i_{\nu-1} \operatorname{rem} t)}$, for each $j \in [t]$. Since $\gamma_{\nu-1}$ is an isomorphism from $\mathcal{N}_{\nu-1}^{\mathfrak{A}}(\bar{a}^{(i_{\nu-1})})$ to $\mathcal{N}_{\nu-1}^{\mathfrak{A}}(\bar{a}^{(i_\nu)})$, we have $\gamma_{\nu-1}(\bar{a}_{(j+i_{\nu-1} \operatorname{rem} t)}) = \bar{a}_{(j+i_\nu \operatorname{rem} t)}$. Altogether, we obtain that ρ_ν maps each element of the tuple $\bar{a}^{(0)}$ to the element in the same position of the tuple $\bar{a}^{(i_\nu)}$, as required. In order to show that (3.2) holds for ν, it remains to show for each relation $R \in \sigma$ of arity r and each tuple $\bar{c} \in A^r$ that $\bar{c} \in R^{\mathfrak{A}_w}$ iff $\rho_\nu(\bar{c}) \in R^{\mathfrak{A}}$. If $\bar{c}$ contains only elements from $N_{\nu-1}^{\mathfrak{A}_w}(\bar{a}^{(0)})$, this follows since $\gamma_{\nu-1} \circ \rho_{\nu-1}$ is an isomorphism. It remains to consider tuples $\bar{c}$ containing elements from $S_\nu^{\mathfrak{A}_w}(\bar{a}^{(0)})$. That is, $\bar{c} \in (S_{\nu-1}^{\mathfrak{A}_w} \cup S_\nu^{\mathfrak{A}_w})^r$. As in the definition of $\mathfrak{A}_w$, we assume for notational simplicity that $\bar{c} = (\bar{c}_{\nu-1}, \bar{c}_\nu)$ where the elements of $\bar{c}_{\nu-1}$ and $\bar{c}_\nu$ belong to $S_{\nu-1}^{\mathfrak{A}_w}$ and $S_\nu^{\mathfrak{A}_w}$, respectively. Here, $\bar{c}_{\nu-1}$ could be the empty tuple. By case (ii) of the definition of $\mathfrak{A}_w$, we see that $\bar{c} \in R^{\mathfrak{A}_w}$ iff $(\bar{c}_{\nu-1}, \pi^{-1}(\bar{c}_\nu)) \in R^A$. Since π is, in particular, a partial isomorphism from $\mathfrak{A}$ to $\mathfrak{A}$ which is defined on all elements of $(\bar{c}_{\nu-1}, \pi^{-1}(\bar{c}_\nu))$, the previous statement holds iff $\pi((\bar{c}_{\nu-1}, \pi^{-1}(\bar{c}_\nu))) = (\pi(\bar{c}_{\nu-1}), \bar{c}_\nu) \in R^A$. Since, as an isomorphism, π preserves distances, the elements of $\pi(\bar{c}_{\nu-1})$ belong to $S^{\mathfrak{A}}_{\nu-1}(\bar{a}^{(i)})$, for each $i \in [t]$. Since $\rho_{\nu-1} \restriction S^{\mathfrak{A}_w}_{\nu-1}(\bar{a}^{(0)}) = \mathrm{id}_{\nu-1}$ and $\rho_\nu \restriction S^{\mathfrak{A}_w}_{\nu}(\bar{a}^{(0)}) = \mathrm{id}_\nu$, we have $\pi(\bar{c}_{\nu-1}) = \gamma_{\nu-1}(\rho_{\nu-1}(\bar{c}_{\nu-1})) = \rho_\nu(\bar{c}_{\nu-1})$ and $\bar{c}_\nu = \rho_\nu(\bar{c}_\nu)$. Hence, $(\pi(\bar{c}_{\nu-1}), \bar{c}_\nu) = (\rho_\nu(\bar{c}_{\nu-1}), \rho_\nu(\bar{c}_\nu)) = \rho_\nu(\bar{c})$ and we are done. □

For the proof of Theorem 3.5.7, we want to convert the circuit $\tilde{C}$ of the previous lemma, without increasing its size too much, to a circuit which accepts exactly the binary words that contain a number of ones which is divisible by r, where $r \geqslant 2$ is some factor of the number t from the previous lemma.

Lemma 3.5.11. *Let $m, d, M, t, p \in \mathbb{N}_{\geqslant 1}$ with $m > 9$ and $p, t \geqslant 2$ such that p and t are coprime. Let $\tilde{C}$ be a MOD_p-circuit of depth d and size M which has the property that for all words $w, w' \in \{0,1\}^m$ with $|w|_1 \equiv |w'|_1 \,(mod\, t)$, it accepts w iff it accepts w'. Furthermore, let $\tilde{C}$ accept all $w \in \{0,1\}^m$ with $|w|_1 \equiv 0 \,(mod\, t)$, and reject all $w \in \{0,1\}^m$ with $|w|_1 \equiv 1 \,(mod\, t)$.*

There is a MOD_p-circuit $\hat{C}$ of depth $(d{+}6)$ and size $(tM{+}2m^t)$ which, for some factor $r \geqslant 2$ of t, accepts exactly those binary words $w \in \{0,1\}^m$ where $|w|_1 \equiv 0 \,(mod\, r)$.

Proof. We let $b = b_0 b_1 \cdots b_{t-1}$ be the binary word of length t where, for every $j \in [t]$ we have $b_j = 1$ iff $\tilde{C}$ accepts binary words $w \in \{0,1\}^m$ with $|w|_1 \equiv j \,(\mathrm{mod}\, t)$.

For a binary word $w \in \{0,1\}^m$ with $|w|_0 \geqslant t{-}1$, we let $pattern(w) = a_0 a_1 \cdots a_{t-1} \in \{0,1\}^t$ with $a_j = 1$ iff $\tilde{C}$ accepts the binary word obtained from w by replacing the first j zeros with ones. Note that if $|w|_1 \equiv i \,(\mathrm{mod}\, t)$, then $pattern(w) = b_i b_{i+1} \cdots b_{t-1} b_0 \cdots b_{i-1}$.

Claim 3.5.12. *There is a factor $r \geqslant 2$ of t such that for all $w \in \{0,1\}^m$ with $|w|_0 \geqslant t{-}1$ we have: $pattern(w) = b \iff |w|_1 \equiv 0 \,(mod\, r)$.*

Proof. Obviously, $pattern(w) = b$ is true for all $w \in \{0,1\}^m$ with $|w|_0 \geqslant t-1$ and $|w|_1 \equiv 0 \,(\mathrm{mod}\, t)$. In case that for all $w \in \{0,1\}^m$ with $|w|_0 \geqslant t-1$ we have

$$pattern(w) = b \iff |w|_1 \equiv 0 \,(\mathrm{mod}\, t),$$

we are done by choosing $r := t$.

In case that there is a $w \in \{0,1\}^m$ with $|w|_0 \geqslant t{-}1$, $pattern(w) = b$, and $|w|_1 \equiv i \,(\mathrm{mod}\, t)$ for an $i \in [1, t{-}1]$, we know that $b_0 b_1 \cdots b_{t-1} = b_i b_{i+1} \cdots b_{t-1} b_0 \cdots b_{i-1}$. Thus, for $x := b_0 \cdots b_{i-1}$ and $y := b_i \cdots b_{t-1}$ we have $b = xy = yx$, and $x, y \in \{0,1\}^+$.

A basic result in word combinatorics (see Proposition 1.3.2 in [Lot84]) states that two words $x, y \in \{0,1\}^+$ commute (i.e., $xy = yx$) iff they are powers of the same word (i.e., there is a $z \in \{0,1\}^+$ and $\nu, \mu \in \mathbb{N}_{\geqslant 1}$ such that $x = z^\nu$ and $y = z^\mu$). We choose $z \in \{0,1\}^+$ of minimal length such that $b = z^s$ for some $s \in \mathbb{N}$. Clearly, $|z| \geqslant 2$, since by assumption we have $b_0 b_1 = 10$.

Since z is of minimal length, it is straightforward to see that for every $w \in \{0,1\}^m$ with $|w|_0 \geqslant t{-}1$ we have: $pattern(w) = z^s \iff |w|_1 \equiv 0 \,(\mathrm{mod}\, |z|)$. □

We choose r according to the claim. Obviously, the following is true for every $w \in \{0,1\}^m$:

$$|w|_1 \equiv 0 \,(\mathrm{mod}\, r) \iff \begin{cases} (1) & |w|_0 \geqslant t{-}1 \text{ and } \mathit{pattern}(w) = b, \text{ or} \\ (2) & \text{there is } j \in [t{-}1] \text{ with } m{-}j \equiv 0 \,(\mathrm{mod}\, r) \text{ such that } |w|_0 = j. \end{cases}$$

To complete the proof of Lemma 3.5.11, it therefore suffices to construct circuits $C_{(1)}$ and $C_{(2)}$ testing for (1) and (2), respectively, and to let $\hat{C}$ be the disjunction of $C_{(1)}$ and $C_{(2)}$. For constructing these circuits, note the following:

- For each $j \leqslant m$, a circuit C_j of depth 2 and size $\leqslant m^j$ which accepts an input $w \in \{0,1\}^m$ iff $|w|_0 = j$ can be realised via $\bigvee_{\substack{I\subseteq[1,m]\\|I|=j}} \big(\bigwedge_{i\in I} \neg w_i \;\wedge\; \bigwedge_{i\in[1,m]\setminus I} w_i\big)$.
- The circuit $C_{(2)}$ can be realised via $\bigvee_{\substack{j\in[t-1]\\ j\equiv 0 \,(\mathrm{mod}\, r)}} C_j$. In particular, $C_{(2)}$ has depth 2 and size $< m^t$.
- A circuit $C_{\geqslant t-1}$ of depth 2 and size $\leqslant m^{t-1}$ which accepts an input $w \in \{0,1\}^m$ iff $|w|_0 \geqslant t{-}1$ can be realised via $\bigvee_{\substack{I\subseteq[1,m]\\|I|=t-1}} \bigwedge_{i\in I} \neg w_i$.
- It is not difficult to construct, for each $j \in [t]$ a circuit $C^{(j)}$ with $2m$ output gates $w_1^{(j)}, \neg w_1^{(j)}, \ldots, w_m^{(j)}, \neg w_m^{(j)}$, such that on input of a $w \in \{0,1\}^m$ with $|w|_0 \geqslant t{-}1$, this circuit produces at its output gates $w_1^{(j)}, \ldots, w_m^{(j)}$ the word $w^{(j)} = w_1^{(j)} \cdots w_m^{(j)}$ obtained from w by replacing the first j zeros with ones. For $j \geqslant 1$ note that $w_i^{(j)}$ can be expressed as
$$w_i \;\vee\; \Big(\neg w_i \;\;\wedge \bigvee_{\substack{I\subseteq[1,i-1]\\|I|<j}} \big(\bigwedge_{i'\in I} \neg w_{i'} \;\;\wedge \bigwedge_{i'\in[1,i-1]\setminus I} w_{i'}\big)\Big).$$
  Thus, the circuit $C^{(j)}$ has depth 4 and size $2m(3 + m^{j-1}) \leqslant 8m^j$.
- For each $j \in [s]$ let $\tilde{C}^{(j)}$ be the concatenation of $C^{(j)}$ and $\tilde{C}$. Note that on input of a binary word $w \in \{0,1\}^m$ with $|w|_0 \geqslant t{-}1$, the circuit $\tilde{C}^{(j)}$ computes, at its output gate, the letter a_j of $\mathit{pattern}(w) = a_0 \cdots a_{t-1}$. Furthermore, $\tilde{C}^{(j)}$ has depth $(4{+}d)$ and size $\leqslant 8m^j + M$.
- The circuit $C_{(1)}$ can now be realised via
$$C_{\geqslant t-1} \;\;\wedge \bigwedge_{\substack{j\in[t]\\ b_j=1}} \tilde{C}^{(j)} \;\;\wedge \bigwedge_{\substack{j\in[t]\\ b_j=0}} \neg\tilde{C}^{(j)}.$$
  This circuit has depth $(d{+}5)$ and size $\leqslant 1 + m^{t-1} + \sum_{j=0}^{t-1}(8m^j + M) \leqslant tM + m^t$ (for $m > 9$).
- Finally, the circuit $\hat{C}$ is realised via $C_{(1)} \vee C_{(2)}$. Thus, $\hat{C}$ has depth $(d{+}6)$ and size $\leqslant tM + 2m^t$.

$\square$

We are now ready for the proof of Theorem 3.5.7.

Proof of Theorem 3.5.7. Let q be a kt-ary query defined on $\mathfrak{C}$ by an $\mathcal{ARB}$-inv-FO+MOD$_p[\sigma]$-formula $\varphi(\overline{x}_0, \dots, \overline{x}_{t-1})$, where $\overline{x}_i$ is a k-tuple of variables, for each $i \in [t]$. By Theorem 3.5.9, there exist numbers $d, s \in \mathbb{N}$ such that for every $n \in \mathbb{N}_{\geqslant 1}$ there is a MOD$_p$-circuit C_n with $\lambda^{\sigma}_{kt}(n)$ inputs, depth d, and size n^s such that the following is true for all σ-structures $\mathfrak{A} \in \mathfrak{C}$ with $|A| = n$, all k-tuples $\overline{a}_0, \dots, \overline{a}_{t-1} \in A^k$, and all embeddings ι of $\mathfrak{A}$:

$$C_n \text{ accepts } \mathrm{Rep}^{\iota}(\mathfrak{A}, \overline{a}_0, \dots, \overline{a}_{t-1}) \iff \mathfrak{A}^{\iota} \models \varphi[\overline{a}_0, \dots, \overline{a}_{t-1}] \iff \mathfrak{A} \models \varphi[\overline{a}_0, \dots, \overline{a}_{t-1}].$$

For contradiction, assume that for every $c \in \mathbb{N}$ the query q is *not* shift $(\log n)^c$-local w.r.t. t on $\mathfrak{C}$. Thus, in particular for $c := 2\ell(d{+}6)$ (with ℓ chosen as in Theorem 3.5.8), we obtain that for all $n_0 \in \mathbb{N}$ there is an $n \geqslant n_0$, and a σ-structure $\mathfrak{A} \in \mathfrak{C}$ with $|A| = n$, and k-tuples $\overline{a}_0, \dots, \overline{a}_{t-1} \in A^k$ such that for $m := (\log n)^c = (\log n)^{2\ell(d+6)}$ we have:

- $\mathcal{N}^{\mathfrak{A}}_m(\overline{a}_i) \cong \mathcal{N}^{\mathfrak{A}}_m(\overline{a}_j)$ and $N^{\mathfrak{A}}_m(\overline{a}_i) \cap N^{\mathfrak{A}}_m(\overline{a}_j) = \emptyset$ for all $i, j \in [t]$ with $i \neq j$, and
- $\mathfrak{A} \models \varphi[\overline{a}_0, \overline{a}_1, \dots, \overline{a}_{t-1}]$ and $\mathfrak{A} \not\models \varphi[\overline{a}_1, \dots, \overline{a}_{t-1}, \overline{a}_0]$.

We fix $n \in \mathbb{N}$ sufficiently large such that, for $\hat{d} := (d{+}6)$ and ε and $m_{\hat{d}}$ chosen as in Theorem 3.5.8, we have for $m = (\log n)^c$ that $m > 9$, $m \geqslant m_{\hat{d}}$, and $n^{\varepsilon \log n} > tn^s + 2(\log n)^{ct}$.

Clearly, C_n is a MOD$_p$-circuit with $\lambda^{\sigma}_{kt}(n)$ inputs which, for every $i \in [t]$ and all embeddings ι_1 and ι_2 of $\mathfrak{A}$ accepts $\mathrm{Rep}^{\iota_1}(\mathfrak{A}, \bar{a}^{(i)})$ iff it accepts $\mathrm{Rep}^{\iota_2}(\mathfrak{A}, \bar{a}^{(i)})$. Furthermore, C_n accepts $\mathrm{Rep}^{\iota}(\mathfrak{A}, \bar{a}^{(0)})$ and rejects $\mathrm{Rep}^{\iota}(\mathfrak{A}, \bar{a}^{(1)})$, for every embedding ι of $\mathfrak{A}$. Thus, from Lemma 3.5.10 we obtain a MOD$_p$-circuit $\tilde{C}$ on m inputs, with depth d and size n^s, such that $\tilde{C}$ has the property that for all words $w, w' \in \{0,1\}^m$ with $|w|_1 \equiv |w'|_1 \pmod{t}$, it accepts w iff it accepts w'. Furthermore, $\tilde{C}$ accepts all $w \in \{0,1\}^m$ with $|w| \equiv 0 \pmod{t}$ and rejects all $w \in \{0,1\}^m$ with $|w| \equiv 1 \pmod{t}$. From Lemma 3.5.11, we therefore obtain a MOD$_p$-circuit $\hat{C}$ of depth $\hat{d} := (d{+}6)$ and size $(tn^s + 2m^t) = (tn^s + 2(\log n)^{ct})$ which, for some factor $r \geqslant 2$ of t, accepts exactly those binary words $w \in \{0,1\}^m$ where $|w|_1 \equiv 0 \pmod{r}$.

Since p and t are coprime by assumption, and $r \geqslant 2$ is a factor of t, we know that r has a prime factor different from p's prime factor. Thus, from Theorem 3.5.8 (for $\varepsilon, \ell, m_{\hat{d}}$ chosen as in Theorem 3.5.8, and for $m \geqslant m_{\hat{d}}$) we know that the size $(tn^s + 2(\log n)^{ct})$ of $\hat{C}$ must be bigger than $2^{\varepsilon \sqrt[\ell\hat{d}]{m}}$. However, we chose $m = (\log n)^c = (\log n)^{2\ell\hat{d}}$, and hence $2^{\varepsilon \sqrt[\ell\hat{d}]{m}} = 2^{\varepsilon \cdot (\log n)^2} = n^{\varepsilon \cdot \log n} > tn^s + 2(\log n)^{ct}$ for all sufficiently large n — a contradiction! Thus, the proof of Theorem 3.5.7 is complete. $\square$

3.5.4 Applications

In the same way as Gaifman locality (cf., e.g., [Lib04]), also shift locality can be used for showing that certain queries are not expressible in particular logics. The first example query we consider here is the reachability query *reach* which associates, with every finite directed graph $\mathfrak{A} = (A, E^{\mathfrak{A}})$, the relation

$$\textit{reach}(\mathfrak{A}) := \{(a,b) : \mathfrak{A} \text{ contains a directed path from node } a \text{ to node } b\}.$$

Proposition 3.5.13. *Let $\sigma = \{E\}$ consist of a binary relation symbol E. Let $p, t \in \mathbb{N}$ with $p, t \geqslant 2$ be such that every t-ary query q definable in $\mathcal{ARB}$-inv-FO+MOD$_p[\sigma]$ is shift $f_q(n)$-local w.r.t. t, for a function $f_q : \mathbb{N} \to \mathbb{N}$ where $f_q(n) \leqslant (\frac{n}{2t} - \frac{1}{2})$ for all sufficiently large n. Then, the reachability query is not definable in $\mathcal{ARB}$-inv-FO+MOD$_p[\sigma]$.*

Proof. Assume, for contradiction, that *reach* is definable by an $\mathcal{ARB}$-inv-FO+MOD$_p[\sigma]$-formula $\varrho(x, y)$. Then, $\psi(x_0, \dots, x_{t-1}) := \varrho(x_0, x_1) \wedge \varrho(x_1, x_2) \wedge \cdots \wedge \varrho(x_{t-2}, x_{t-1})$ is an $\mathcal{ARB}$-inv-FO+MOD$_p[\sigma]$-formula expressing in a finite directed graph $\mathfrak{A}$, that there is a directed path from node x_i to node x_{i+1}, for every $i \in [t{-}1]$. Let q be the t-ary query defined by $\psi(x_0, \dots, x_{t-1})$. By assumption, this query is shift $f_q(n)$-local w.r.t. t, for a function f_q with $f_q(n) \leqslant \frac{n}{2t} - \frac{1}{2}$ for all sufficiently large n.

Now, consider for each $\ell \in \mathbb{N}_{\geqslant 1}$ the graph $\mathfrak{A}_\ell$ consisting of a single directed path $v_1 \to v_2 \to \cdots \to v_{t(2\ell+1)}$ on $t{\cdot}(2\ell{+}1)$ nodes. For each $i \in [t]$ let $a_i := v_{i(2\ell+1)+(\ell+1)}$. Then, the ℓ-spheres of the a_i, for $i \in [t]$, are pairwise disjoint and isomorphic. The cardinality of $\mathfrak{A}_\ell$ is $n := t{\cdot}(2\ell{+}1)$, and thus $\ell = \frac{n}{2t} - \frac{1}{2} \geqslant f_q(n)$. Since q is shift $f_q(n)$-local w.r.t. t, we obtain that $\mathfrak{A}_\ell \models \psi[a_0, a_1, \dots, a_{t-1}] \iff \mathfrak{A}_\ell \models \psi[a_1, \dots, a_{t-1}, a_0]$. But in $\mathfrak{A}_\ell$ there is a directed path from a_i to a_{i+1} for every $i \in [t{-}1]$, but there is no directed path from a_{t-1} to a_0. According to the choice of ψ, we have that $\mathfrak{A}_\ell \models \psi[a_0, a_1, \dots, a_{t-1}]$ but $\mathfrak{A}_\ell \not\models \psi[a_1, \dots, a_{t-1}, a_0]$ — a contradiction! □

As an immediate consequence of Proposition 3.5.13 and Theorem 3.5.7 we obtain (the known fact) that the reachability query is not definable in $\mathcal{ARB}$-inv-FO+MOD$_p[E]$, for any prime power p. Proposition 3.5.13 also shows why it can be expected to be difficult to generalise Theorem 3.5.5 and Theorem 3.5.7 from prime powers p to arbitrary numbers $p \geqslant 2$.

Remark 3.5.14. Assume, we could generalise Theorem 3.5.7 from prime powers p to arbitrary numbers $p \geqslant 2$. By Proposition 3.5.13, we would then obtain that the reachability query is not definable in $\mathcal{ARB}$-inv-FO+MOD$_p[E]$, for any $p \in \mathbb{N}$ with $p \geqslant 2$. The "opposite direction" of Theorem 3.5.9, obtained in [BIS90], would then tell us that the reachability query is not computable in ACC^0. Here, ACC$^0 = \bigcup_{p \geqslant 2} \mathrm{AC}^0[p]$, where AC$^0[p]$ is the class of all problems computable by a family of constant depth, polynomial size MOD$_p$-circuits. Since the reachability query can be computed in nondeterministic logarithmic space, we would thus obtain that NLOGSPACE $\not\subseteq$ ACC^0. This would constitute a major breakthrough in computational complexity: The current state-of-the-art (see [Wil11] for a recent survey) states that NEXP $\not\subseteq$ ACC^0, but does not know a problem in P that provably does not belong to ACC^0.

Similarly, a generalisation of Theorem 3.5.5 to all odd numbers p would imply that the reachability query is not definable in AC$^0[p]$, for any odd number p. Also this is currently not known.

The fact that FO-definable queries are Gaifman local simplifies and unifies many non-expressbility results for FO. In fact, to someone acquainted with Gaifman locality, the proof of a non-expressibility result by a locality argument can often be communicated in a simple visual way, while the complicated combinatorial arguments in game based non-expressibility results are hidden within the proof of Gaifman locality for FO. We give some further examples of well-known queries which have already been used in a similar context in the literature on locality and we show that shift locality can also be used to obtain non-expressibility of these queries in $\mathcal{ARB}$-inv-FO+MOD$_p[E]$, for prime powers p. For none of these examples, the fact that they are not expressible in $\mathcal{ARB}$-inv-FO+MOD$_p[E]$, for any prime power p, is new. That is, all examples could be proved directly using Smolensky's theorem. But we believe that they show that the notion of shift locality serves a similar purpose as Gaifman locality. That is, the locality argument simplifies and unifies the proofs, i.e. all details of the necessary reductions are done in Theorem 3.5.7.

Using similar constructions as in the proof of Proposition 3.5.13, we show that none of the following queries is definable in $\mathcal{ARB}$-inv-FO+MOD$_p[E]$, for any prime power p:

- *cycle*$(\mathfrak{A}) := \{a \in A : a$ is a node that lies on a cycle of the graph $\mathfrak{A}\}$,
- *triangle-reach*$(\mathfrak{A}) := \{a \in A : a$ is reachable from a triangle in the graph $\mathfrak{A}\}$,
- *same-distance*$(\mathfrak{A}) := \{(a,b,c) \in A^3 : \operatorname{dist}^{\mathfrak{A}}(a,c) = \operatorname{dist}^{\mathfrak{A}}(b,c)\}$.

In the literature, variants of these queries have previously served as examples of queries which are not Gaifman local. In the proofs below, we will use the following notion. Given a directed graph $\mathfrak{A} = (A, E^{\mathfrak{A}})$ and an edge $e = (u,v) \in E^{\mathfrak{A}}$, the *$\ell$-fold subdivision* of (u,v) replaces the edge (u,v) by a path $u \to u_{e,1} \to \cdots \to u_{e,\ell} \to v$, where $u_{e,1}, \ldots, u_{e,\ell} \notin A$ are new nodes introduced for subdividing e. This notion extends naturally to edge sets instead of single edges: The ℓ-fold subdivision of a set E of directed edges is obtained by replacing each edge $(u,v) \in E$ by a path $u \to u_{e,1} \to \cdots \to u_{e,\ell} \to v$, where $u_{e,1}, \ldots, u_{e,\ell}$ are new nodes introduced for subdividing the edge e.

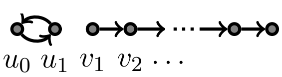
Proposition 3.5.15.
For any prime power p, the cycle query is not definable in $\mathcal{ARB}$-inv-FO+MOD$_p[E]$.

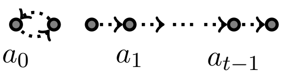
Proof. Let $t \in \mathbb{N}$ with $t \geqslant 2$ such that t and p are coprime (e.g., $t = 2$ if p is odd, and $t = 3$ if p is even). Assume, for contradiction, that the *cycle* query is definable by an $\mathcal{ARB}$-inv-FO+MOD$_p[E]$-formula $\varrho(x)$. Let $x_0, \ldots, x_{t-1}$ be first-order variables that do not occur in $\varrho(x)$ and let

$$\psi(x_0, \ldots, x_{t-1}) \;:=\; \varrho(x_0) \;\wedge \bigwedge_{1 \leqslant i < t} x_i{=}x_i.$$

Let q be the t-ary query defined by $\psi(x_0, \ldots, x_{t-1})$. By Theorem 3.5.7, the query q is shift $(\log n)^c$-local, for some $c \in \mathbb{N}$.

Let $\mathfrak{A}_0$ be a directed graph which is a disjoint union of a directed cycle $u_0 \to u_1 \to u_0$ and a directed path $v_0 \to v_1 \to \cdots \to v_t$.





For each $\ell \in \mathbb{N}_{\geqslant 1}$, let $\mathfrak{A}_\ell$ be obtained by the ℓ-fold subdivision of the edge set of $\mathfrak{A}_0$. In particular, $\mathfrak{A}_\ell$ has cardinality $n_\ell = t{+}3{+}(t{+}2)\ell$. Let $a_0 := u_0$, and for each $i \in [1, t{-}1]$, let $a_i := v_i$. For sufficiently large values of ℓ, the $(\log n_\ell)^c$-spheres of the nodes $a_0, \ldots, a_{t-1}$ in $\mathfrak{A}_\ell$ are pairwise disjoint and isomorphic, each of them being isomorphic to a directed path of length $2(\log n_\ell)^c{+}1$ with node a_i in the middle.

Since q is shift $(\log n)^c$-local w.r.t. t, we obtain for all sufficiently large ℓ that

$$\mathfrak{A}_\ell \models \psi[a_0, a_1, \ldots, a_{t-1}] \iff \mathfrak{A}_\ell \models \psi[a_1, \ldots, a_{t-1}, a_0].$$

However, $a_0 \in \textit{cycle}(\mathfrak{A}_\ell)$, because a_0 lies on a cycle, and $a_1 \notin \textit{cycle}(\mathfrak{A}_\ell)$, because a_1 does not lie on a cycle. That is, $\mathfrak{A}_\ell \models \varrho[a_0]$ and $\mathfrak{A}_\ell \not\models \varrho[a_1]$. Thus, according to the choice of the formula ψ, we have that $\mathfrak{A}_\ell \models \psi[a_0, a_1, \ldots, a_{t-1}]$, but $\mathfrak{A}_\ell \not\models \psi[a_1, \ldots, a_{t-1}, a_0]$, which is a contradiction. □

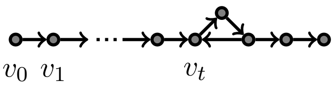
Proposition 3.5.16.
For any prime power p, the triangle-reach query is not definable in $\mathcal{ARB}$-inv-FO+MOD$_p[E]$.

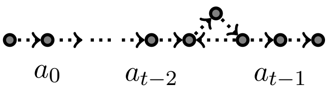
Proof. Let $t \in \mathbb{N}$ with $t \geqslant 2$ such that t and p are coprime. Assume, for contradiction, that the *triangle-reach* query is definable by an $\mathcal{ARB}$-inv-FO+MOD$_p[E]$-formula $\varrho(x)$. Let $x_0, \ldots, x_{t-1}$ be first-order variables that do not occur in $\varrho(x)$ and let

$$\psi(x_0, \ldots, x_{t-1}) \;:=\; \varrho(x_{t-1}) \,\wedge \bigwedge_{0 \leqslant i < t-1} x_i{=}x_i.$$

Let q be the t-ary query defined by $\psi(x_0, \ldots, x_{t-1})$. By Theorem 3.5.7, the query q is shift $(\log n)^c$-local, for some $c \in \mathbb{N}$.

Let $\mathfrak{A}_0$ be the graph consisting of the directed path $v_0 \to \cdots \to v_t$, the directed triangle $v_t \to v_{t+1} \to v_{t+2} \to v_t$, and the directed path $v_{t+2} \to v_{t+3} \to v_{t+4}$.





For each $\ell \in \mathbb{N}_{\geqslant 1}$, let $\mathfrak{A}_\ell$ be obtained by the ℓ-fold subdivision of all but the triangle's edges of $\mathfrak{A}_0$. In particular, $\mathfrak{A}_\ell$ has cardinality $n_\ell = t{+}5 + (t{+}2)\ell$. For each $i \in [t{-}1]$, let $a_i := v_{i+1}$, and let $a_{t-1} := v_{t+3}$. For sufficiently large values of ℓ, the $(\log n_\ell)^c$-spheres of the nodes $a_0, \ldots, a_{t-1}$ in $\mathfrak{A}_\ell$ are pairwise disjoint and isomorphic, each of them being isomorphic to a directed path of length $2(\log n_\ell)^c{+}1$ with node a_i in the middle.

Since q is shift $(\log n)^c$-local w.r.t. t, we obtain for all sufficiently large ℓ that

$$\mathfrak{A}_\ell \models \psi[a_0, a_1, \ldots, a_{t-1}] \iff \mathfrak{A}_\ell \models \psi[a_1, \ldots, a_{t-1}, a_0].$$

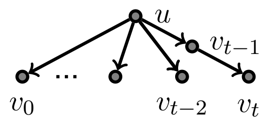
However, $a_{t-1} \in \textit{triangle-reach}(\mathfrak{A}_\ell)$, because a_{t-1} is reachable from the triangle, and $a_0 \notin \textit{triangle-reach}(\mathfrak{A}_\ell)$, because there is no directed path from any node of the triangle to a_0. That is, $\mathfrak{A}_\ell \models \varrho[a_{t-1}]$ and $\mathfrak{A}_\ell \not\models \varrho[a_0]$. Thus, due to the choice of the formula ψ, we have that $\mathfrak{A}_\ell \models \psi[a_0, a_1, \ldots, a_{t-1}]$, but $\mathfrak{A}_\ell \not\models \psi[a_1, \ldots, a_{t-1}, a_0]$, which is a contradiction. □

Proposition 3.5.17.
For any prime power p, the same-distance query is not definable in $\mathcal{ARB}$-inv-FO+MOD$_p[E]$.

Proof. Let $t \in \mathbb{N}$ with $t \geqslant 3$ such that t and p are coprime. Assume, for contradiction, that the *same-distance* query is definable by an $\mathcal{ARB}$-inv-FO+MOD$_p[E]$-formula $\varrho(x, y, z)$. Let $x_0, \ldots, x_{t-1}$ be first-order variables that do not occur in $\varrho(x, y, z)$ and let

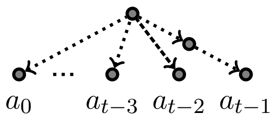
$$\psi(x_0, \ldots, x_{t-1}) \;:=\; \varrho(x_{t-3}, x_{t-2}, x_{t-1}) \;\wedge \bigwedge_{0 \leqslant i < t-3} x_i{=}x_i.$$

Let q be the t-ary query defined by $\psi(x_0, \ldots, x_{t-1})$. By Theorem 3.5.7, the query q is shift $(\log n)^c$-local, for some $c \in \mathbb{N}$.



Let $\mathfrak{A}_0$ be the graph consisting of the edges $u \to v_i$ for each $i \in [t{-}1]$, and a directed path $u \to v_{t-1} \to v_t$.

For each $\ell \in \mathbb{N}_{\geqslant 1}$, let $\mathfrak{A}_\ell$ be the graph obtained by the ℓ-fold subdivision of the edges $u \to v_i$ in $\mathfrak{A}_0$, for each $i \in [t]$.



In particular, $\mathfrak{A}_\ell$ has cardinality $n_\ell = t{+}2 + t\ell$. For each $i \in [t{-}1]$ let $a_i := v_i$, and let $a_{t-1} := v_t$. For sufficiently large values of ℓ, the $(\log n_\ell)^c$-spheres of nodes the $a_0, \ldots, a_{t-1}$ in $\mathfrak{A}_\ell$ are pairwise disjoint and isomorphic, each of them being isomorphic to a directed path of length $(\log n_\ell)^c{+}1$ with a_i at the end of the path.

Since q is shift $(\log n)^c$-local w.r.t. t, we obtain for all sufficiently large ℓ that

$$\mathfrak{A}_\ell \models \psi[a_0, a_1, \ldots, a_{t-1}] \iff \mathfrak{A}_\ell \models \psi[a_1, \ldots, a_{t-1}, a_0].$$

However, $(a_{t-3}, a_{t-2}, a_{t-1}) \in$ *same-distance*$(\mathfrak{A}_\ell)$, because a_{t-3} and a_{t-2} both have distance $2(\ell{+}1){+}1$ to a_{t-1} in $\mathfrak{A}_\ell$, and $(a_{t-2}, a_{t-1}, a_0) \notin$ *same-distance*$(\mathfrak{A}_\ell)$, because a_{t-2} has distance $2(\ell{+}1)$ to a_0, but a_{t-1} has distance $2(\ell{+}1){+}1$ to a_0. That is, $\mathfrak{A}_\ell \models \varrho[a_{t-3}, a_{t-2}, a_{t-1}]$ and $\mathfrak{A}_\ell \not\models \varrho[a_{t-2}, a_{t-1}, a_0]$. Thus, due to the choice of the formula ψ, we have that $\mathfrak{A}_\ell \models \psi[a_0, a_1, \ldots, a_{t-1}]$, but $\mathfrak{A}_\ell \not\models \psi[a_1, \ldots, a_{t-1}, a_0]$, which is a contradiction. $\square$

3.6 Hanf locality and locality on words

For giving the precise definition of Hanf locality, we need the following notation: As in [Lib04], for σ-structures $\mathfrak{A}$ and $\mathfrak{B}$, for k-tuples $\overline{a} \in A^k$ and $\overline{b} \in B^k$, and for an $r \in \mathbb{N}$, we write $(\mathfrak{A}, \overline{a}) \leftrightarrows_r (\mathfrak{B}, \overline{b})$ (or simply $\mathfrak{A} \leftrightarrows_r \mathfrak{B}$ in case that $k{=}0$) if there is a bijection $\beta : A \to B$ such that $\mathcal{N}_r^{\mathfrak{A}}(\overline{a}c) \cong \mathcal{N}_r^{\mathfrak{B}}(\overline{b}\beta(c))$ is true for every $c \in A$.

Definition 3.6.1 (Hanf locality)**.** Let $\mathfrak{C}$ be a class of finite σ-structures, $k \in \mathbb{N}$, and $f : \mathbb{N} \to \mathbb{N}$.
A k-ary query q is *Hanf $f(n)$-local on* $\mathfrak{C}$ if there is an $n_0 \in \mathbb{N}$ such that for every $n \in \mathbb{N}$ with $n \geqslant n_0$ and all σ-structures $\mathfrak{A}, \mathfrak{B} \in \mathfrak{C}$ with $|A| = |B| = n$, the following is true for all k-tuples $\overline{a} \in A^k$ and $\overline{b} \in B^k$ with $(\mathfrak{A}, \overline{a}) \leftrightarrows_{f(n)} (\mathfrak{B}, \overline{b})$: $\quad \overline{a} \in q(\mathfrak{A}) \iff \overline{b} \in q(\mathfrak{B})$.
The query q is called *Hanf $f(n)$-local* if it is Hanf $f(n)$-local on the class of all finite σ-structures.

Consider a 0-ary query which maps a structure, containing a unary relation P, to the relation containing only the empty tuple if the number of elements contained in P is even, and to the empty set, otherwise. This is an example of a Hanf 0-local query. The 0-ary query which maps a graph to the relation containing only the empty tuple if it is connected, and to the empty set otherwise, is an example of a query which is not Hanf $f(n)$-local for any sublinear function f.

Hanf locality is an even stronger locality notion than Gaifman locality:

Theorem 3.6.2 (Hella, Libkin, Nurmonen [HLN99]). *Let $\mathfrak{C}$ be a class of finite σ-structures and let $f : \mathbb{N} \to \mathbb{N}$. Let $k \in \mathbb{N}_{\geqslant 1}$ and let q be a k-ary query. If q is Hanf $f(n)$-local on $\mathfrak{C}$, then q is Gaifman $(3f(n){+}1)$-local on $\mathfrak{C}$.*

It is well-known that queries definable in FO or FO+MOD$_p$ (for any $p \geqslant 2$) are Hanf local with a constant locality radius [FSV95, HLN99]. For order-invariant or $\mathcal{ARB}$-invariant FO it is still open whether they are Hanf local with respect to any sublinear locality radius. As an immediate consequence of Proposition 3.5.2 and Theorem 3.6.2 one obtains for every $p \in \mathbb{N}$ with $p \geqslant 2$ that order-invariant FO+MOD$_p$ is *not* Hanf local with respect to any sublinear locality radius.

Recall that an $\mathcal{ARB}$-inv-FO+MOD$_p[\sigma_\Sigma]$-sentence φ defines the language $L_\varphi := \{w \in \Sigma^+ : w \models \varphi\}$. Furthermore, a language $L \subseteq \Sigma^+$ corresponds to a 0-ary query q_L on Σ^+ defined, for each $w \in \Sigma^+$, via

$$q_L(w) \ := \ \begin{cases} \{()\} & \text{if } w \in L \\ \emptyset & \text{if } w \notin L, \end{cases}$$

where $()$ denotes the empty tuple. A language $L \subseteq \Sigma^+$ is called *Hanf $f(n)$-local* iff the 0-ary query q_L is Hanf $f(n)$-local on Σ^+, i.e., there is an $n_0 \in \mathbb{N}$ such that for every $n \in \mathbb{N}$ with $n \geqslant n_0$ and all words $u, v \in \Sigma^+$ of length n with $\mathfrak{S}_u \leftrightarrows_{f(n)} v$, we have: $u \in L \iff v \in L$.

For the restricted case of word structures, Benedikt and Segoufin [BS09b] have shown that on Σ^+ order-invariant FO has the same expressive power as FO and thus is Hanf local with constant locality radius (in fact, [BS09b] obtains the same result also for finite labelled ranked trees). In [AvMSS12] it was shown that every query definable in $\mathcal{ARB}$-invariant FO on Σ^+ is Hanf local with polylogarithmic locality radius, and that in the worst case the locality radius can indeed be of polylogarithmic size. As an immediate consequence of Proposition 3.5.4 and Theorem 3.6.2 we obtain that for $\Sigma := \{0,1\}$ there is a unary query q that is *not* Hanf $(\frac{n-8}{12})$-local on Σ^+, but definable in $<$-inv-FO+MOD$_p[\sigma_\Sigma]$ for every *even* number $p \geqslant 2$.

From this observation, we can also obtain an example of a language which is not Hanf $(\frac{n-8}{12})$-local on words over an extended alphabet. The existence of this language can be obtained using a general principle which allows to convert a k-ary query q over words to a 0-ary query, i.e. a language A_q over words over an extended alphabet. The language A_q inherits the relevant definability and locality properties for our purposes from the query q. This principle is stated in Lemma 3.6.9 below, since we will need it for the proof of Theorem 3.6.5 below. Using this approach here, however, it becomes rather hard to describe the shape of the language which is not Hanf $(\frac{n-8}{12})$-local concretely. To this end, we show directly how this language can be obtained by simple modification of the proof of Proposition 3.5.4. Consider the languages

$$\begin{aligned} L_{\mathit{left}} &:= 1^+\,2\,0^+1^+0^+ \\ L_{\mathit{right}} &:= 1^+0^+1^+\,2\,0^+ \end{aligned}$$

over the alphabet $\Sigma := \{0,1,2\}$. Note that the definitions of L_{left} and L_{right} are very similar, the only difference being the position of the unique 2 occurring in words from L_{left} and L_{right}. We define

$$\begin{aligned} L_{\mathit{even}} &:= \{w \in L_{\mathit{right}} : |w|_1 \,(\mathrm{mod}\, 2) \equiv 0\} \\ L_{\mathit{odd}} &:= \{w \in L_{\mathit{left}} : |w|_1 \,(\mathrm{mod}\, 2) \equiv 1\} \end{aligned}$$

Proposition 3.6.3. *Let* $L := L_{\mathit{even}} \cup L_{\mathit{odd}}$.

1. *The language* L *is not Hanf* $(\frac{n-1}{8})$*-local.*
2. L *is definable by a sentence* φ *in* $<$-inv-FO+MOD$_p[\sigma_\Sigma]$*, for every even number* $p \geqslant 2$.

Proof. Ad 1: For every $\ell \in \mathbb{N}_{\geqslant 1}$ let

$$\begin{aligned} u_\ell &:= 1^\ell\ 1^\ell 2 0^\ell\ 0^\ell 1^\ell\ 1^\ell 0^\ell\ 0^\ell &= x\, y_1\, y_2\, y_3\, z, \\ v_\ell &:= 1^\ell\ 1^\ell 0^\ell\ 0^\ell 1^\ell\ 1^\ell 2 0^\ell\ 0^\ell &= x\, y_3\, y_2\, y_1\, z, \end{aligned}$$

for $\quad x := 1^\ell, \quad y_1 := 1^\ell 2 0^\ell, \quad y_2 := 0^\ell 1^\ell, \quad y_3 := 1^\ell 0^\ell, \quad z := 0^\ell.$

It is not difficult to see that $u_\ell \leftrightarrows_\ell v_\ell$: the bijection β, for which

$$\mathcal{N}_\ell^{u_\ell}(c) \;\cong\; \mathcal{N}_\ell^{v_\ell}(\beta(c)) \qquad \text{for every } \ c \in [|u_\ell|],$$

can be chosen as follows. It maps each position of

- x in u_ℓ onto the according position of x in v_ℓ,
- y_s (for $s \in \{1,2,3\}$) in u_ℓ onto the according position of y_s in v_ℓ,
- z in u_ℓ onto the according position of z in v_ℓ.

It is straightforward to verify that this bijection β indeed witnesses that $u_\ell \leftrightarrows_\ell v_\ell$. Furthermore, $v_\ell \in L$ and $u_\ell \notin L$. The length of u_ℓ and v_ℓ is $n := 8\ell{+}1$, thus $\ell = \frac{n-1}{8}$. Hence, the language L is not Hanf $(\frac{n-1}{8})$-local. This completes the proof of *(a)*.

Ad 2: First, note that the language

$$M := L_{\mathit{left}} \cup L_{\mathit{right}}$$

can be defined by an FO$[\sigma_\Sigma]$-sentence φ_M which states the following:

- The first position of the word carries the letter 1. The last position of the word carries the letter 0.
- For each position x that carries the letter 1, the position directly to the right of x carries one of the letters 0, 1, 2. Furthermore, there is exactly one position x that carries the letter 1, such that the position directly to the right of x carries the letter 0. And there is exactly one position x that carries the letter 1, such that the position directly to the right of x carries the letter 2.
- For each position y that carries the letter 0, the position directly to the right of y carries one of the letters 0, 1. Furthermore, there is exactly one position y that carries the letter 0, such that the position directly to the right of y carries the letter 1.
- There is exactly one position z that carries the letter 2. The position directly to the right of z carries the letter 0.

From Example 3.4.1 we obtain an $<$-inv-FO+MOD$_2[E]$-sentence $\varphi_{\textit{even cycles}}$ that is satisfied by a finite $\{E\}$-structure $\mathfrak{A}$ iff $\mathfrak{A}$ is a disjoint union of directed cycles where the number of cycles of even length is even. We choose

$$\varphi \ := \ \varphi_M \wedge \varphi',$$

where φ' is the formula obtained from $\varphi_{\textit{even cycles}}$ by relativisation of all quantifiers to the positions that carry the letters 1 or 2, and by replacing every atom of the form $E(\mu, \nu)$ (for first-order variables μ and ν) by a formula stating that

- $E(\mu, \nu)$ is true, or
- position μ carries the letter 2, and ν is the leftmost position of the word, or
- the positions μ and ν both carry the letter 1, and the positions directly to the right of μ and directly to the left of ν both carry the letter 0.

Clearly, the obtained formula is order-invariant on the class of all finite σ_Σ-structures, since the formula $\varphi_{\textit{even cycles}}$ is order-invariant on the class of all finite $\{E\}$-structures.

It is straightforward to see that, when evaluated in a word $w \in L_{\textit{left}}$ of the form $1^i 2 0^j 1^{i'} 0^{j'}$, the formula φ' simulates the evaluation of the formula $\varphi_{\textit{even cycles}}$ in a graph that consists of the disjoint union of two cycles of lengths $i{+}1$ and i'; and hence φ' is satisfied iff $i{+}1$ and i' are either both even or both odd — and this is equivalent to the statement that $|w|_1 = i{+}i'$ is odd. Similarly, when evaluated in a word $w \in L_{\textit{right}}$ of the form $1^i 0^j 1^{i'} 2 0^{j'}$, the formula φ' simulates the evaluation of the formula $\varphi_{\textit{even cycles}}$ in a graph that consists of a single cycle of length $i{+}i'{+}1$; and hence φ' is satisfied iff $i{+}i'{+}1$ is odd, i.e., $|w|_1 = i{+}i'$ is even.

In summary, we obtain that φ is an $<$-inv-FO+MOD$_2[\sigma_\Sigma]$-sentence that defines the language L. Since modulo 2 counting quantifiers can be simulated by modulo p counting quantifiers, for every *even* number $p \geqslant 2$, the proof of (b) is complete. $\square$

Remark 3.6.4. Benedikt and Segoufin [BS09b] conjectured that order-invariant FO+MOD (i.e. formulae with arbitrary modulo-counting quantifiers) has the same expressive power as FO+MOD on trees. The previous proposition shows that, for even numbers $p \geqslant 2$, there exist order-invariant FO+MOD$_p$-definable languages which are not FO+MOD-definable, since FO+MOD is Hanf local. This refutes the conjecture even for words instead of trees.

From Niemistö's Corollary 7.25 in [Nie07] it follows that for *odd* numbers p, order-invariant FO+MOD$_p[\sigma_\Sigma]$ on Σ^+ has the same expressive power as FO+MOD$_{\mathrm{PFC}(p)}[\sigma_\Sigma]$, where PFC$(p)$ is the set of all numbers whose prime factors are prime factors of p, and by FO+MOD$_{\mathrm{PFC}(p)}$ we denote first-order logic with modulo m counting quantifiers for all $m \in \mathrm{PFC}(p)$.

The present section's main result shows that for *odd prime powers* p, the Hanf locality result of [AvMSS12] can be generalised from $\mathcal{ARB}$-invariant FO to $\mathcal{ARB}$-invariant FO+MOD$_p$ on Σ^+:

Theorem 3.6.5. *Let Σ be a finite alphabet. Let $k \in \mathbb{N}$, let q be a k-ary query, and let p be an odd prime power. If q is definable in $\mathcal{ARB}$-inv-FO+MOD$_p[\sigma_\Sigma]$ on Σ^+, then there is a $c \in \mathbb{N}$ such that q is Hanf $(\log n)^c$-local on Σ^+.*

Together with Theorem 3.6.2 this implies (general instead of weak) Gaifman locality on Σ^+:

Corollary 3.6.6. *Let Σ be a finite alphabet. Let $k \in \mathbb{N}_{\geqslant 1}$, let q be a k-ary query, and let p be an odd prime power. If q is definable in $\mathcal{ARB}$-inv-FO+MOD$_p[\sigma_\Sigma]$ on Σ^+, then there is a $c \in \mathbb{N}$ such that q is Gaifman $(\log n)^c$-local on Σ^+.*

Note that this corollary does not contradict the non-locality result of Proposition 3.5.2, as the counter-example given in the proof of that proposition is not a word structure.

The remainder of this section is devoted to the proof of Theorem 3.6.5. We follow the overall approach of [AvMSS12]. The crucial step is to prove Theorem 3.6.5 for queries q of arity $k = 0$. The case for queries of arity $k \geqslant 1$ can then easily be reduced to the case for queries of arity 0 by adding k extra symbols to the alphabet; see below.

Note that a 0-ary query q defines the language $L_q := \{w \in \Sigma^+ : () \in q(w)\}$, where $()$ denotes empty tuple. The language L_q is called *Hanf $f(n)$-local* iff q is Hanf $f(n)$-local on Σ^+. For proving Theorem 3.6.5 for the case $k = 0$, we consider the following notion.

Definition 3.6.7 (Disjoint swaps [AvMSS12])**.** Let $r \in \mathbb{N}$ and let $w \in \Sigma^+$ be a word over a finite alphabet Σ. A word $w' \in \Sigma^+$ is obtained from w by a *disjoint r-swap operation* if there exist words x, u, y, v, z such that $w = xuyvz$ and $w' = xvyuz$, and for the positions i, j, i', j' of w just before u, y, v, z the following is true: The neighbourhoods $N_r^w(i)$, $N_r^w(j)$, $N_r^w(i')$, $N_r^w(j')$ are pairwise disjoint, and $\mathcal{N}_r^w(i) \cong \mathcal{N}_r^w(i')$ and $\mathcal{N}_r^w(j) \cong \mathcal{N}_r^w(j')$.

Let $f : \mathbb{N} \to \mathbb{N}$. A language $L \subseteq \Sigma^+$ is *closed under disjoint $f(n)$-swaps* if there exists an $n_0 \in \mathbb{N}$ such that for every $n \in \mathbb{N}_{\geqslant 1}$ with $n \geqslant n_0$, all words $w \in \Sigma^+$ of length n, and all words w' obtained from w by a disjoint $f(n)$-swap operation, we have: $w \in L \iff w' \in L$.

Note that these swaps are a special case of the vertical swaps on trees which we will study in Chapter 4 (cf. Section 4.4.1), but the disjointness condition is slightly more

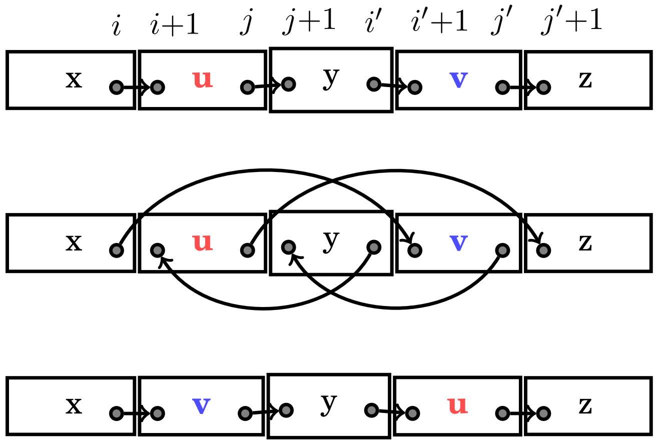


Figure 3.6.1: The picture shows how the disjoint r-swap operation turns the edge relation of the word structure of a word $w = xuyvz$ (top) into the edge relation of the word structure of $w' = xvyuz$ (middle and bottom).

restrictive than the most restrictive condition studied there (strong guardedness, cf. Section 4.6).

It was shown in [AvMSS12] (see Proposition 5.7, Lemma 5.2, and the proof of Theorem 5.1 in [AvMSS12]) that if a language $L \subseteq \Sigma^+$ is closed under disjoint $(\log n)^d$-swaps, for some $d \in \mathbb{N}$, then it is Hanf $(\log n)^c$-local, for some $c > d$. Hence, the following lemma immediately implies Theorem 3.6.5 for the case $k = 0$.

Lemma 3.6.8. *Let Σ be a finite alphabet, let $L \subseteq \Sigma^+$, and let p be an odd prime power. If L is definable by an $\mathcal{ARB}$-inv-FO+MOD$_p[\sigma_\Sigma]$-sentence, then there exists a constant $d \in \mathbb{N}$ such that L is closed under disjoint $(\log n)^d$-swaps.*

Proof. We proceed in the same way as in the proof of Proposition 5.5 in [AvMSS12], which obtained the analogue of Lemma 3.6.8 for $\mathcal{ARB}$-inv-FO$[\sigma_\Sigma]$-sentences. However, we cannot just copy the proof from there, since that proof relies on (general) Gaifman locality (with polylogarithmic locality radius) of queries definable in $\mathcal{ARB}$-inv-FO$[\sigma_\Sigma]$, while in the present case we have available (from Theorem 3.5.5) only *weak* Gaifman locality (with polylogarithmic locality radius) of queries definable in $\mathcal{ARB}$-inv-FO+MOD$_p[\sigma_\Sigma]$.

Let φ be an $\mathcal{ARB}$-inv-FO+MOD$_p[\sigma_\Sigma]$-sentence which defines a word language $L_\varphi = \{w \in \Sigma^+ : w \models \varphi\}$. For contradiction, assume that there is no $d \in \mathbb{N}$ such that L_φ is closed under disjoint $(\log n)^d$-swaps. For each fixed $d, n_0 \in \mathbb{N}$ let $n \geqslant n_0$ and let w, w' be words of length n which witness the violation of the "closure under disjoint $(\log n)^d$-swaps" property. That is, $w \in L_\varphi$, $w' \notin L_\varphi$, and w' is obtained from w by a disjoint $(\log n)^d$-swap operation. Thus, there exist words x, u, y, v, z such that $w = xuyvz$ and $w' = xvyuz$, and for the positions i, j, i', j' of w just before u, y, v, z the following is true for $r := (\log n)^d$: the neighbourhoods $N_r^w(i)$, $N_r^w(j)$, $N_r^w(i')$, $N_r^w(j')$ are pairwise disjoint, and $\mathcal{N}_r^w(i) \cong \mathcal{N}_r^w(i')$ and $\mathcal{N}_r^w(j) \cong \mathcal{N}_r^w(j')$.

The overall proof idea is as follows:

1. Choose an appropriate extension $\tilde{\sigma}$ of the signature σ_Σ,
2. modify the formula φ into a suitable FO+MOD$_p(\tilde{\sigma}_{\mathcal{ARB}})$-formula $\psi(x_1)$ with one free variable, and
3. define for each word w and all tuples $\overline{p} := (i, i', j, j')$ of positions of w which satisfy $0 \leqslant i < i' < j < j' < |w|$ a $\tilde{\sigma}$-structure $\mathfrak{A} := \mathfrak{A}_w^{\overline{p}}$ with the same universe as w, such that the positions $a := i{+}1$ and $a' := i'{+}1$ have disjoint and isomorphic $((\log n)^d{-}1)$-spheres in $\mathfrak{A}$ if the $(\log n)^d$-spheres of i and i' and those of j and j' in w are isomorphic and all these $(\log n)^d$-spheres are pairwise disjoint

such that the following is satisfied:

4. $\psi(x_1)$ is $\mathcal{ARB}$-invariant on $\mathfrak{A}$,
5. $\mathfrak{A} \models \psi[a] \iff w \models \varphi$,
6. $\mathfrak{A} \models \psi[a'] \iff w' \models \varphi$.

Note that $w \in L_\varphi$ and $w' \notin L_\varphi$ imply that $\mathfrak{A} \models \psi[a]$ and $\mathfrak{A} \not\models \psi[a']$. In combination with (3) this shows that the unary query defined by $\psi(x_1)$ is *not* weakly Gaifman $((\log n)^d{-}1)$-local on the class $\mathfrak{C}$ containing all structures $\mathfrak{A}_w^{\overline{p}}$ which we defined above. Property (4) is true for each choice of the word w and positions $\overline{p}$. Hence, $\psi(x_1)$ is $\mathcal{ARB}$-invariant on $\mathfrak{C}$. Since d can be chosen arbitrarily large, and our choice of the formula $\psi(x_1)$ will not depend on d, this contradicts the fact that the formula $\psi(x_1)$ is weakly Gaifman local on $\mathfrak{C}$ with polylogarithmic locality radius according to Theorem 3.5.5.

The details described in items (1)–(4) are carried out as follows.

ad (1): Let $\tilde{\sigma} := \sigma_\Sigma \cup \{F, X, Y_1, Y_2, Z\}$, where F is a binary relation symbol and X, Y_1, Y_2, Z are unary relation symbols. Thus, $\tilde{\sigma} = \{E, F, X, Y_1, Y_2, Z\} \cup \{P_a : a \in \Sigma\}$.

ad (3): Let $\mathfrak{A} := \mathfrak{A}_w^{\overline{p}}$ be the $\tilde{\sigma}$-structure defined as follows (an illustration can be found in Figure 3.6.2):

- $\mathfrak{A}$ has the same universe as w, i.e., $A = [|w|]$.
- $E^{\mathfrak{A}}$ is obtained from E^w by removing all edges between the words x, u, y, v, z, i.e. $E^{\mathfrak{A}} \ := \ E^w \setminus \{(i, i{+}1),\ (j, j{+}1),\ (i', i'{+}1),\ (j', j'{+}1)\}$.
- $F^{\mathfrak{A}}$ relates the first and last position of u, and the first and last position of v, i.e. $F^{\mathfrak{A}} \ := \ \{(i{+}1, j),\ (i'{+}1, j')\}$.
- $X^{\mathfrak{A}}$ marks the last position of x, $Y_1^{\mathfrak{A}}$ and $Y_2^{\mathfrak{A}}$ mark the first and the last position of y, and $Z^{\mathfrak{A}}$ marks the first position of z, i.e. $X^{\mathfrak{A}} := \{i\}$, $Y_1^{\mathfrak{A}} := \{j{+}1\}$, $Y_2^{\mathfrak{A}} := \{i'\}$, $Z^{\mathfrak{A}} := \{j'{+}1\}$.
- for each $a \in \Sigma$, $P_a^{\mathfrak{A}}$ is identical to P_a^w, i.e. $P_a^{\mathfrak{A}} = P_a^w$.

It is easy to see that, if in w the $(\log n)^d$-spheres of i and i' are isomorphic and the $(\log n)^d$-spheres of j and j' are isomorphic and the $(\log n)^d$-spheres of i, i', j, j' are pairwise disjoint, then in $\mathfrak{A}$ the $((\log n)^d{-}1)$-spheres of $a := i{+}1$ and $a' := i'{+}1$ are disjoint and isomorphic.

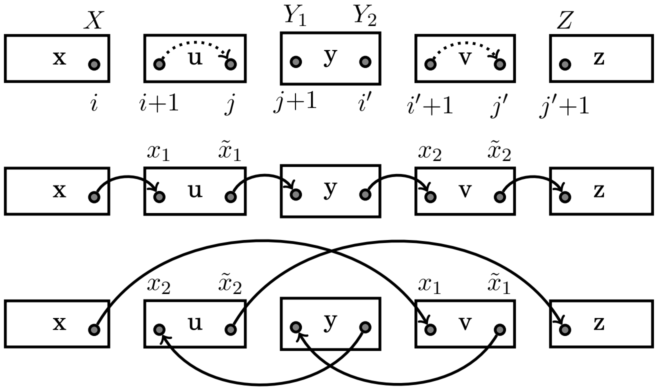


Figure 3.6.2: Structure $\mathfrak{A}$ (top) of Lemma 3.6.8 and the edge relations simulated by the formula $\psi(x_1)$ if x_1 is assigned the value $a := i{+}1$ (middle) or the value $a' := i'{+}1$ (bottom). F-edges in $\mathfrak{A}$ are depicted by dotted arcs; the E-edges in $\mathfrak{A}$ are given as successor relations within each of the framed boxes.

ad (2) and (4), (5), (6): We define $\psi(x_1)$ in such a way that, when evaluated in one of the structures $\mathfrak{A} := \mathfrak{A}_w^{\overline{p}}$, it does the following:

- if the variable x_1 is assigned the value $a := i{+}1$, then $\psi(x_1)$ simulates φ on w .
- if the variable x_1 is assigned the value $a' := i'{+}1$, then $\psi(x_1)$ simulates φ on w'
- if x_1 is assigned to a value different from a, a', then $\psi(x_1)$ is not satisfied in $\mathfrak{A}$.

The first two items imply items (5) and (6) above. Together with the last item and the $\mathcal{ARB}$-invariance of φ on w and w', this implies that item (4) is satisfied.
We let $\psi(x_1) := \exists\tilde{x}_1\exists x_2\exists\tilde{x}_2\ \psi'$, where ψ' is a conjunction of formulae stating that:

- $F(x_1, \tilde{x}_1)$ holds. (This ensures that $\psi(x_1)$ is satisfied in $\mathfrak{A}$ only if x_1 is assigned one of the values $a := i{+}1$ or $a' := i'{+}1$.)
- $x_2 \neq x_1$ and $F(x_2, \tilde{x}_2)$ holds.
- The formula $\varphi'(x_1, \tilde{x}_1, x_2, \tilde{x}_2)$ is satisfied, where φ' is obtained from φ by replacing each atom of the form $E(\mu, \nu)$ with the formula $\chi_E(x_1, \tilde{x}_1, x_2, \tilde{x}_2, \mu, \nu)$ defined as follows:
  χ_E is the disjunction of formulae stating that
  - $E(\mu, \nu)$ holds (i.e., there already exists an E-edge from μ to ν in $\mathfrak{A}$),

- $X(\mu)$ holds and $\nu = x_1$ (i.e., a new E-edge from the last position of the word x to the position assigned to the variable x_1 is introduced)
- $\mu = \tilde{x}_1$ and $Y_1(\nu)$ holds (i.e., a new E-edge from the position assigned to $\tilde{x}_1$ to the first position of the word y is introduced)
- $Y_2(\mu)$ and $\nu = x_2$ (i.e., a new E-edge from the last position of the word y to the position assigned to the variable x_2 is introduced)
- $\mu = \tilde{x}_2$ and $Z(\nu)$ (i.e., a new E-edge from the position assigned to the variable $\tilde{x}_2$ to the first position of the word z is introduced).

It is not difficult to see that χ_E simulates the edge relation of w in $\mathfrak{A}$ if the variable x_1 is assigned the value $a := i{+}1$; and it simulates the edge relation of w' if the variable x_1 is assigned the value $a' := i'{+}1$ (see Figure 3.6.2 for an illustration).

This finishes the construction for items (2) and (4), (5), (6). In summary, the proof of Lemma 3.6.8 is complete. □

Now that we have proved Theorem 3.6.5 for the case $k = 0$, we explain how the general case can be obtained from this. To this end, we convert a k-ary query on words to a language over an extended alphabet with the same relevant definability and locality properties. This can be done using a standard technique which encodes variable assignments in an extended alphabet.

For each alphabet Σ and each $k \geqslant 1$, we let $\Sigma_{\text{var}(k)} := \Sigma \times 2^{[k]}$. The subsets of $[k]$ are used to encode an assignment of k variables to the positions of a word over the alphabet Σ. We let $L_{\text{assign}(k)}$ denote the language of words $w \in \Sigma^+_{\text{var}(k)}$ where for each number $i \in [k]$ there is a unique position $\text{occ}_i(w) \in [|w|]$ such that the label of $\text{occ}_i(w)$ is $(a, X) \in \Sigma_{\text{var}(k)}$ with $i \in X$. Note that this language is clearly FO$[\sigma_{\Sigma_{\text{var}(k)}}]$-definable. For each word $w \in L_{\text{assign}(k)}$, we let $\tilde{w} \in \Sigma^+$ denote the word where each symbol (a, X) is replaced by a. Furthermore, we let $\overline{\text{occ}_k}(w) := (\text{occ}_0(w), \ldots, \text{occ}_{k-1}(w))$.

Lemma 3.6.9. *Let Σ be a finite alphabet and let $k \in \mathbb{N}_{\geqslant 1}$. With each k-ary query q on Σ^+ we associate the language*

$$A_q \;:=\; \{w \in L_{\text{assign}(k)} : \overline{\text{occ}}_k(w) \in q(S_{\tilde{w}})\}.$$

Then

1. *For each $p \in \mathbb{N}_{\geqslant 1}$, if q is definable in $\mathcal{ARB}$-*inv-FO+MOD$_p[\sigma_\Sigma]$ *on Σ^+, then A_q is definable in $\mathcal{ARB}$-*inv-FO+MOD$_p[\sigma_{\Sigma_{\text{var}(k)}}]$.
2. *The query q is Hanf $f(n)$-local on Σ^+ iff the language A_q is Hanf $f(n)$-local.*

Proof.

Ad (1): Let $\varphi(\overline{x})$ be a formula of $\mathcal{ARB}$-inv-FO+MOD$_p[\sigma_\Sigma]$ defining q, where $\overline{x} := (x_0, \ldots, x_{k-1})$. For each $i \in [k]$, let $\psi_{\text{occ},i}(x) := \bigvee_{a \in \Sigma, X \in 2^{[k]}, i \in X} P_{(a,X)}(x)$. That is, the formula states that i occurs in the set in the second component of the label at position x. Let

$$\hat{\varphi} := \exists x_0 \cdots \exists x_{k-1} \bigwedge_{i \in [k]} \psi_{\text{occ},i}(x_i) \wedge \varphi',$$

where φ' is obtained from φ by replacing each occurrence of a relation symbol $P_a(x)$, for $a \in \Sigma$, by the formula $\bigvee_{X \in 2^{[k]}} P_{(a,X)}(x)$ which states for a position x of $\mathfrak{S}_w$, for each word $w \in \Sigma^+_{\text{var}(k)}$, that the label at position x in $\tilde{w}$ is a. Let ψ be an FO-sentence which defines $L_{\text{assign}(k)}$ on $\Sigma^+_{\text{var}(k)}$. It is straightforward to verify that $\hat{\varphi} \wedge \psi$ is a formula of $\mathcal{ARB}$-inv-FO+MOD$_p[\sigma_{\Sigma_{\text{var}(k)}}]$ which defines the language A_q.

Ad (2): Note that $\mathfrak{S}_u \leftrightarrows_{f(n)} \mathfrak{S}_v$ iff $(\mathfrak{S}_{\tilde{u}}, \overline{\text{occ}}_k(u)) \leftrightarrows_{f(n)} (\mathfrak{S}_{\tilde{v}}, \overline{\text{occ}}_k(v))$, for all $u, v \in L_{\text{assign}(k)}$ of length n. More concretely, a bijection β witnesses the first statement iff it witnesses the second statement. This follows from the fact that β preserves the second component of the labelling of $\mathfrak{S}_u$ (i.e. the label of $\beta(x)$ is the same as the label of x, for all positions $x \in [n]$) iff $\beta(\text{occ}_i(u)) = \text{occ}_i(v)$, for all $i \in [k]$, and that the label of each position of $\tilde{u}$ and $\tilde{v}$ is the first component of the labelling of that position in u and v.

Note also that $L_{\text{assign}(k)}$ is 0-local since its definition depends only on the labelling.

First, we show that locality of q implies locality of A_q. Suppose that $\mathfrak{S}_u \leftrightarrows_{f(n)} \mathfrak{S}_v$, for $u, v \in \Sigma^n_{\text{var}(k)}$. We have $u \in L_{\text{assign}(k)}$ iff $v \in L_{\text{assign}(k)}$. If $u, v \notin L_{\text{assign}(k)}$, then $u, v \notin A_q$. Suppose that $u, v \in L_{\text{assign}(k)}$. Then $(\mathfrak{S}_{\tilde{u}}, \overline{\text{occ}}_k(u)) \leftrightarrows_{f(n)} (\mathfrak{S}_{\tilde{v}}, \overline{\text{occ}}_k(v))$ and we obtain that $\overline{\text{occ}}_k(u) \in q(\mathfrak{S}_{\tilde{u}})$ iff $\overline{\text{occ}}_k(v) \in q(\mathfrak{S}_{\tilde{v}})$, by $f(n)$-locality of q, and so $u \in A_q$ iff $v \in A_q$.

Now we show that locality of A_q implies locality of q. Let $u, v \in \Sigma^n$ and $\overline{a} = (a_0, \ldots, a_{k-1}), \overline{b} := (b_0, \ldots, b_{k-1}) \in [n]^k$ be such that $(\mathfrak{S}_u, \overline{a}) \leftrightarrows_{f(n)} (\mathfrak{S}_v, \overline{b})$. There exist $u', v' \in L_{\text{assign}(k)}$ such that $u = \tilde{u}'$ and $v = \tilde{v}'$ and such that $a_i = \text{occ}_i(u')$ and $b_i = \text{occ}_i(v')$, for each $i \in [k]$. Then $(\mathfrak{S}_u, \overline{a}) = (\mathfrak{S}_{\tilde{u}'}, \overline{\text{occ}}_k(u'))$ and $(\mathfrak{S}_v, \overline{b}) = (\mathfrak{S}_{\tilde{v}'}, \overline{\text{occ}}_k(v'))$. Since $(\mathfrak{S}_u, \overline{a}) \leftrightarrows_{f(n)} (\mathfrak{S}_v, \overline{b})$, we also have $\mathfrak{S}_{u'} \leftrightarrows_{f(n)} S_{v'}$. By Hanf $f(n)$-locality of A_q, this means that $u' \in A_q$ iff $v' \in A_q$. Hence, $\overline{\text{occ}}_k(u') \in q(\mathfrak{S}_{\tilde{u}'})$ iff $\overline{\text{occ}}_k(v') \in q(\mathfrak{S}_{\tilde{v}'})$. That is, $\overline{a} \in q(\mathfrak{S}_u)$ iff $\overline{b} \in q(\mathfrak{S}_v)$ and hence q is Hanf $f(n)$-local on Σ^+.

$\square$

The proof of Theorem 3.6.5 can now be finished as follows. Suppose that there is a k-ary query q, for some $k > 0$, which is definable in $\mathcal{ARB}$-inv-FO+MOD$_p[\sigma_\Sigma]$ on Σ^+, for some odd prime power p and some finite alphabet Σ, such that there is no $c \in \mathbb{N}$ for which q is Hanf $(\log n)^c$-local on Σ^+. By the previous lemma, we obtain a language A_q which is definable in $\mathcal{ARB}$-inv-FO+MOD$_p[\sigma_{\Sigma_{\text{var}(k)}}]$ and which is also not Hanf $(\log n)^c$-local. But we have already proved that this is impossible.

3.7 Conclusion

We have introduced a new notion of locality called *shift locality* which generalises the notion of alternating locality with constant locality radius of Niemistö [Nie07] and Libkin's notion of weak Gaifman locality [Lib04]. We have presented a comprehensive picture of the locality of $\mathcal{ARB}$-inv-FO+MOD$_p$ for prime powers p.

We have also investigated notions of locality on word structures. Here some natural questions remain open. We have shown that there is an $<$-inv-FO+MOD$_2$-definable language which is not Hanf local and hence not FO+MOD$_q$-definable for any modulus q. It would be interesting to understand the expressive power of $<$-inv-FO+MOD$_2$ on words in more detail. Is there a decidable algebraic characterisation of the $<$-inv-FO+MOD$_2$-definable languages? Is there a logic with the same expressive power as $<$-inv-FO+MOD$_2$ but with an effective syntax? (A similar question was raised in [Nie07].)

We have derived Hanf locality on words from weak Gaifman locality. To accomplish this, we have used the characterisation of Hanf locality in terms of disjoint swap operations. The origin of these swap operations goes back to [BS09a] where similar operations are, among other things, used to obtain a new proof of an algebraic characterisation of first-order logic on words from [BP89]. As a first step towards proving an algebraic characterisation of $<$-inv-FO+MOD$_2$, we believe that one could define a variant of Hanf locality whose relation to shift locality is similar to the relation of Hanf locality and weak Gaifman locality that we established. Then one could try to characterise this notion of locality by a variant of the disjoint swap operations.

Addition-invariant FO and Regular Tree Languages

In this chapter, we consider the expressive power of addition-invariant FO-sentences ($+$-inv-FO). The main goal of this chapter is to obtain an effective syntax for the $+$-inv-FO-definable regular tree languages.

4.1 Introduction

We start with an introduction, where we first give a short overview of what is known about addition-invariance. This is followed by a discussion of decidable and algebraic characterisations of logics on words and trees, since some of the central tools in this section come from this direction. Afterwards, we discuss the contributions of this chapter.

4.1.1 Addition-invariance

The study of addition-invariant sentences has been initiated in [Sch06], where a natural question about the expressive power of addition-invariant MSO-formulae was shown to be closely related to questions about the strictness of two complexity theoretic hierarchies. Note that it is known (cf. e.g. [Tho96]) that *monadic least fixed-point logic* (MLFP) (i.e. the restriction of least fixed-point logic where only unary relation variables are allowed) and MSO have the same expressive power on finite words. In [Sch06], it was shown that resolving the questions about the relationship of $+$-inv-MLFP and $+$-inv-MSO on finite words would have severe complexity theoretic consequences. If $+$-inv-MLFP and $+$-inv-MSO have the same expressive power, then $\text{PH} = \text{P}$. In other words, the *polynomial time hierarchy* collapses to deterministic polynomial time and, in particular, $\text{P} = \text{NP}$. If not, then $\text{DLIN} \neq \text{LINH}$. That is, the *linear time hierarchy* (a linear time analogue to the polynomial time hierarchy) is separated from deterministic linear time (cf. [Gra93] for the definition of these complexity classes). While, arguably, this makes the second alternative more plausible, it can also be seen as evidence that understanding the expressive power of $+$-inv-MSO on finite words is difficult. As a first step, it seems reasonable to try to understand the expressive power of $+$-inv-FO on words. The paper of Schweikardt and Segoufin [SS10] has obtained first results towards this goal. They obtained an effective syntax for the *regular word languages* definable by $+$-inv-FO-sentences, both for the representation of words as successor structures and as linearly ordered structures. More concretely, they showed that a regular word language is $+$-inv-FO-definable iff it is definable in the logic $\mathrm{FO}_{\mathrm{card}}$ which extends FO by *cardinality predicates* that, for each $m \in \mathbb{N}^+$, allow to express that the length of a word can be divided by m. Among other things, the proof used an *algebraic characterisation* of the $\mathrm{FO}_{\mathrm{card}}$-definable tree languages. That is, the paper found properties of the *syntactic monoid* of a regular language such that the language is $\mathrm{FO}_{\mathrm{card}}$-definable iff it satisfies these properties. Since the properties of the syntactic monoid are decidable and the syntactic monoid can be computed for any given automaton, this result yields a *decidable characterisation* of the $\mathrm{FO}_{\mathrm{card}}$-definable and the $+$-inv-FO-definable regular tree languages, i.e. there is an algorithm which, on input of an automaton, decides whether the language recognised by the automaton defines an $\mathrm{FO}_{\mathrm{card}}$-definable (and hence an $+$-inv-FO-definable language) language.

Our goal is to obtain an effective syntax for the $+$-inv-FO-definable *regular tree languages*. To this end, we extend the mentioned algebraic and decidable characterisation obtained by [SS10] from words to trees.

4.1.2 Algebraic and decidable characterisations of logics on words and trees

The search for decidable characterisations of certain classes of regular languages has a long tradition in automata theory. The goal is to find an algorithm which decides, on input of an automaton, if the language recognised by the automaton belongs to a fixed class of regular languages. In particular, it can be asked if there exist decidable characterisations for the class of regular languages which are definable by any given logic. For the case of *word languages* many instances of this problem have been solved by obtaining *algebraic characterisations* of the language classes in terms of properties of the syntactic monoids of the languages. The fundamental theorem of Büchi [Büc60] establishes a link between automata theory and logic by showing that a word language is MSO-definable iff it is regular. It turned out that many fragments of MSO correspond to classes of regular languages which have a decidable algebraic characterisation. A detailed overview of these matters can be found in [Str94]. For example, the regular languages definable by FO over word structures with the natural linear order (i.e. the transitive closure of the successor relation in a word structure) are exactly the languages whose syntactic monoids are *aperiodic* (cf. [Sch65, MP71]) and the regular languages definable by FO on word structures using only the successor relations and no linear order are precisely the languages with aperiodic syntactic monoids that satisfy certain identities which allow to swap the position of two factors in a product, provided that they are surrounded by the same idempotent elements of the monoid (cf. [BP89]). This property has been called *closure under idempotent-guarded swaps* in [BS09a]. Similar results are known for extensions of FO such as FO+MOD (cf. [Str94]). Another example is the characterisation of the logic $\mathrm{FO}_{\mathrm{card}}$ of [SS10] mentioned above. According to this, a regular language is $\mathrm{FO}_{\mathrm{card}}$-definable over word structures with the natural linear order if it satisfies a property which is called *closure under transfer* [SS10], and a regular language is $\mathrm{FO}_{\mathrm{card}}$-definable over word structures without a linear order iff it is *closed under transfer* and closed under idempotent-guarded swaps.

Transferring such characterisations from word languages to *tree languages* is usually quite a challenge. Recall that a rooted and labelled finite tree is usually represented as a finite relational structure with unary relations for the labelling of the nodes and binary relations for its successor relations. Often the *prefix order*, i.e. the transitive closure of the parent-child relation, is also added to this representation.

It is a longstanding open problem to find a decidable characterisation of the regular tree languages definable by FO with prefix order. This problem has been investigated, at least, since the beginning of the 1990s (cf. e.g. [Pot94]). One major obstacle is that it is not quite clear what kind of algebraic object should replace the syntactic monoids which are used in the decidable characterisations of languages of words. In recent years, some decidable characterisation for fragments of FO on trees/forests with prefix order have been obtained using the *forest algebra* formalism (cf. e.g. [BSS12]). But the problem to characterise full FO remains open.

From now on, we restrict our discussion to FO with successor relations but without the prefix order. Here, the situation looks differently. It has been known for a long time that these languages are exactly the *locally threshold testable* languages (cf. e.g [Tho96]). Roughly, this characterisation which can be derived from Hanf's theorem (cf. [Han65], [FSV95]) states that, without prefix order, each FO-sentence can only count the number of occurrences of subtrees of a constant height k up to a constant threshold. This is a useful characterisation for many purposes, but it yields no decidable characterisation of the FO-definable languages. A decidable characterisation was announced by Benedikt and Segoufin in [BS05]. It was stated in terms of a generalisation of closure under idempotent-guarded swaps from word languages to tree languages. Unfortunately, this characterisation turned out to be wrong, but the same authors later successfully obtained a different decidable characterisation of FO in the paper [BS09a]. To this end, they introduced two new operations on trees which, together, are called *guarded swaps*. These operations combine the idea of the operations of the previously suggested characterisation with the idea of locally threshold testability. Whereas previously the operations allowed to swap the positions of two parts of a tree as long as they are surrounded by trees which induce the same idempotent transition function, now two parts of a tree could be exchanged as long as they are *k-similar*, i.e. the trees of height at most k which separate the parts from the surrounding tree must be identical. If a language is locally threshold testable, then it must be *closed under guarded swaps*, i.e. there must be a k such that applying these operations to a tree does not lead out of the language. The paper [BS09a] showed that, together with the requirement that the language be regular and aperiodic, closure under guarded swaps characterises the FO-definable tree languages. They also showed that this characterisation yields that FO-definability is decidable.

The guarded swap operations and the characterisation have seen several other related applications. For instance, Place and Segoufin [PS11] used the swap operations together with another operation to characterise a subclass of the locally threshold testable tree languages, the *locally testable tree languages*. Anderson, van Melkebeek, Schweikardt, and Segoufin [AvMSS12] used the same operation with a somewhat different notion of guardedness to characterise the *Hanf local* (cf. [HLN99]) word languages. Benedikt and

Segoufin [BS09b] applied their characterisation of the FO-definable tree languages in their result that $<$-inv-FO and FO have the same expressive power on trees. Guarded swaps play a central role in the following chapter.

4.2 Contributions

We discuss the main results of this chapter and their relation to previous results in the literature. The results of this chapter are based on work which has first been announced in the conference paper [HS12]. The full results contained in this chapter have not been previously published. The presentation in this thesis reuses some parts of [HS12].

All results concern languages of ranked, finite, labelled trees with successor relations and without prefix order.

A decidable characterisation of $\mathbf{FO_{card}}$ on trees

For our *first main result*, we generalise the notion of *closure under transfer* to tree languages, and we show that the FO_{card}-definable tree languages coincide with the regular tree languages that are closed under transfer and under guarded swaps. We show that closure under transfer of a regular tree language is decidable, and hence it is decidable whether a regular tree language is FO_{card}-definable or not.

A decidable syntax for $+$-inv-FO-definable regular tree languages

Our *second main result* shows that the $+$-inv-FO-definable regular tree languages coincide with the FO_{card}-definable tree languages. For the proof of this result, we use our characterisation of FO_{card}-definability and a third technical main result.

A well-behaved notion of guardedness for swaps

Our *third main result* shows that closure under guarded swaps is equivalent to closure under a much more restricted form of swaps which we call *strongly guarded swaps*. This result eradicates many combinatorial complications associated with guarded swaps. It is crucial for the proof of our second main result. Apart from this, it also clarifies some aspects of several earlier results from [BS09b], [BS09a].

4.3 Preliminaries

We assume basic knowledge of automata theory and its relations to semigroup theory (cf. [Str94]) and of tree automata (cf. [CDG$^+$07]). As in all parts of this thesis, words, trees, and logical structures are assumed to be finite. The only exception will occur in Section 4.7 where we consider extensions of the infinite structure $(\mathbb{N}, +)$.

Trees.

There are several different ways to represent *labelled rooted binary trees* with distinguished left and right successor relations formally. One useful possibility (cf. e.g. [Tho96]) is to identify each node with a word over the alphabet $\{0, 1\}$ which describes the path from the root to that node. Here, 0 and 1 represent left and right turns on the path. A labelled tree is then represented as a partial mapping from $\{0, 1\}^*$ to a finite alphabet. This map should assign a label to each node of the tree which translates to the requirement that the domain of the map must be a prefix closed language.

These considerations lead to the following definitions for r-ary trees with r distinguished successors. We let $\Pi_r := \{0, \ldots, r-1\}$ be the *path alphabet* and we let Σ be a finite *label alphabet*. A *Σ-labelled tree of rank k* is then defined as a function $t : \mathrm{dom}(t) \to \Sigma$ whose *domain* $\mathrm{dom}(t) \subseteq \Pi_r^*$ is a prefix closed language. The elements of $\mathrm{dom}(t)$ are called *nodes*. If Σ and r are fixed or unimportant, as in most parts of this chapter, we write Π for Π_r and instead of "Σ-labelled tree of rank r", we say *tree*. A *tree language* is a set of trees. For $u, v \in \Pi^*$ and $i \in \Pi$, if $v = ui$ we say that v is the *i-successor* of u and that u is the *parent* of v. Note that, in a tree, we allow for nodes which have an $(i + 1)$-successor, but no i-successor. A node without successors in a tree t is a *leaf* of t. The *size* of a tree t is $|t| := |\mathrm{dom}(t)|$. The *height* of a tree t is defined as usual as the maximum length of a path from the root to a leaf v of t, i.e. the maximum of $|v|$ for all leaves. The empty word ϵ is the *root* of t. Recall that $\trianglelefteq$ denotes the prefix order on words. If $u \trianglelefteq v$ we say that v lies *below* u, and if $u \triangleleft v$ we say that v lies *strictly below* u. If $u \parallel v$, we also say that u and v are *parallel*. Sometimes we have to choose some linear order on a tree. A canonical choice is the *lexicographic order* on Π^*. Here $u \in \Pi^*$ is *lexicographically smaller* than $v \in \Pi^*$ if there are $x, y, y' \in \Pi^*$ and $i, j \in \Pi$ such that $u = xiy$ and $v = xjy'$ and $i < j$. Observe that the restriction of the lexicographic order to the domain of a tree corresponds to a *breadth-first search order* of the tree.

Logic and trees.

A Σ-labelled tree t of rank r is identified with a logical structure $\mathfrak{A}_t$ over the universe $\mathrm{dom}(t)$ and over a signature $\sigma_{\Sigma,r}$ which contains unary relation symbols P_a, for each $a \in \Sigma$, used to represent the labelling and binary relation symbols S_i, for each $i \in \Pi$, used to represent the i-successor relation. That is, $P_a^{\mathfrak{A}_T} := \{v \in \mathrm{dom}(t) : t(v) = a\}$ and $\mathrm{S}_i^{\mathfrak{A}_T} := \{(u, v) \in \mathrm{dom}(t)^2 : v = ui\}$. Note that we do not include the prefix order as a relation of the structure. Note also that $\mathfrak{A}_s$ and $\mathfrak{A}_t$ are isomorphic iff $s = t$. Most of the time, we do not distinguish between the tree t and the structure $\mathfrak{A}_t$. A language of Σ-labelled trees of rank r is FO-*definable* if there is a first-order-sentence φ over the signature $\sigma_{\Sigma,r}$ such that $t \in L$ iff $t \models \varphi$, for all trees t. Analogous definitions are used for other logics. We identify words with trees of rank 1.

The *distance* $\mathrm{dist}(u, v)$ of $u, v \in \Pi^*$ is defined as their distance in $\mathfrak{A}_t$, for any tree t with $u, v \in \mathrm{dom}(t)$. Note that this definition does not depend on t.

Decompositions of trees.

An *n-context* $C = (t, h_1, \dots, h_n)$, for $n \in \mathbb{N}$, is a tree t with distinguished leaves $h_1, \dots, h_n$. We call t the *underlying tree* of C and $h_1, \dots, h_n$ the *holes* of C. The *inner tree* of C is the tree obtained from t by restriction to $\mathrm{dom}(t) \setminus \{h_1, \dots, h_n\}$. The *size* $|C|$ of C is the size of its inner tree. We identify contexts without holes with their underlying trees. A 1-context is also called *context*, for short.

Given a tree t, some leaf h of t, and another tree t' we can form a new tree by *substituting* the tree t' for the leave h. This yields a tree which we denote by $t[h \hookleftarrow t']$ with $\mathrm{dom}(t[h \hookleftarrow t']) = \mathrm{dom}(t) \cup h\,\mathrm{dom}(t')$ and $t[h \hookleftarrow t'](x) := t(x)$ if $x \in \mathrm{dom}(t) \setminus \{h\}$ and $t[h \hookleftarrow t'](x) := t'(y)$ if $x = hy$. This definition can be extended to n- and m-contexts $C = (t, h_1, \dots, h_n)$ and $D = (t', h'_1, \dots, h'_m)$. Then $C[h_i \hookleftarrow D]$ is the $(m+n-1)$-context with the underlying tree $t[h_i \hookleftarrow t']$ and the holes $h_1, \dots, h_{i-1}, h_{i+1}, \dots, h_n$ and $h_i h'_1, \dots, h_i h'_m$.

We write $C[D_1, \dots, D_n]$ for the tree where each hole h_i is substituted by a context D_i in the way described above. The order of the individual substitution steps is unimportant. If C is a 1-context, we also write $C \cdot D$ or CD for the 1-context $C[D]$. This is the *concatenation* of C with D.

We use contexts to decompose trees. For a tree t with a node u and nodes $v_1, \dots, v_n \trianglerighteq u$, let $t[u, v_1, \dots, v_n)$ be the n-context obtained from the subtree of t rooted at u by removing all nodes strictly below $v_1, \dots, v_n$ and making $v_1, \dots, v_n$ holes.

We often encounter the situation that we are given nodes $u \trianglelefteq v$ of a tree t and we decompose t into contexts $C := t[\epsilon, u)$, $D := t[u, v)$ and a tree $s := t_{|v}$ such that $t = CDs$. As usual, given such a decomposition of t we will consider C, D, s as parts of t. The nodes of C, D, s correspond naturally to nodes of t. For instance, the corresponding node for a node x in D is the node ux of t. It is not necessary to make this correspondence more precise most of the time.

k-types.

Our notation of this section follows the definitions of [BS09a] with minor differences and extensions. Let t be a tree. The *subtree rooted at a node* v is denoted by $t_{|v}$. The *k-spill* or *k-type* of v in t, denoted by $t^k_{|v}$, is the restriction of $t_{|v}$ to all nodes with distance at most k from v. More generally, a k-type is a tree of height at most k. We say that $u \in \mathrm{dom}(t)$ *realises the k-type τ in t* if $t^k_{|u} = \tau$. In this case, we also say that u is an *occurrence* of τ in t or simply that u *has the k-type* τ in t. If it is easily understood to which tree t we refer, we will not mention t. Nodes which have the same k-type are *k-similar*. Trees are k-similar if they have k-similar roots.

For each k-tree τ, $|t|_\tau$ denotes the number of occurrences of τ in the tree t. For a tree s, we write $s \leqslant_k t$ if $|s|_\tau \leqslant |t|_\tau$ holds for all k-types τ. The notation $s =_k t$ and $s <_k t$ is defined analogously. These notations are extended to finite sequences $(t_i)_{i \in [1,n]}$ of trees by the definition $|(t_i)_{i \in [1,n]}|_\tau = |t_1|_\tau + \dots + |t_n|_\tau$. All notations can also be extended to contexts via their underlying trees.

Tree languages and monoids.

Let L be a *tree language*, i.e. a set of trees. Two trees are s and t are *L-equivalent* if either $s, t \in L$ or $s, t \notin L$. Two contexts C_1, C_2 are *L-equivalent*, written $C_1 \cong_L C_2$, if the trees $C \cdot C_1 \cdot t$ and $C \cdot C_2 \cdot t$ are L-equivalent for all contexts C and trees t. A context C is *idempotent* (with respect to L) if $C^2 \cong_L C$. A tree language is *regular*, if it is recognised by a *(deterministic bottom-up) tree automaton*. The set of all contexts with the concatenation operation forms a monoid. The quotient of this monoid by $\cong_L$ is called, in analogy to the algebraic theory of word languages, the *syntactic monoid* of L. Just as in the word case, it is well-known that a tree language is regular iff its syntactic monoid is finite. Therefore, with each regular tree language L come two associated constants: ω_L is the least number such that for each context C, the context C^{ω_L} is idempotent; κ_L is the least number such that for each non-empty context C there exists a non-empty context C' of size $|C'| \in [1, \kappa_L]$ with $C' \cong_L C$. In both cases, we usually omit the index L.

4.4 First-order logic with cardinality predicates

We investigate the expressive power of *first-order logic with cardinality predicates* which allow to count the number of elements in a structure modulo any positive integer. This extension of first-order logic was introduced in [SS10] where it was studied on words. There a decidable characterisation of the languages of words which are definable in this logic was obtained. We study the expressive power of the logic on trees and obtain a decidable characterisation of the definable tree languages. Throughout this section, we fix a finite label alphabet Σ and rank r and consider only Σ-labelled trees of rank r. All logical formulae are over the signature $\sigma_{\Sigma,r}$.

Cardinality predicates

A *cardinality predicate* is a nullary relation symbol $\mathrm{C}_{a,m}$. The atomic formula $\mathrm{C}_{a,m}$ is satisfied in a structure iff the cardinality of the structure's universe is congruent a modulo m. We write $\mathrm{FO}_{\mathrm{card}}$ for the set of all first-order formulae which, besides equality and the relation symbols of the signature for trees, may use atomic formulae $\mathrm{C}_{a,m}$, for each $a \in [m]$ and each $m \in \mathbb{N}^+$.

Our aim is a decidable characterisation of the $\mathrm{FO}_{\mathrm{card}}$-definable tree languages in terms of their closure properties. To this end, we introduce certain operations on trees and we show that a tree language is $\mathrm{FO}_{\mathrm{card}}$-definable iff it is closed under these operations. Then we show that there is an algorithm that decides for a given tree automaton if the language has the right closure properties and hence whether it is $\mathrm{FO}_{\mathrm{card}}$-definable.

4.4.1 Operations on trees: swaps and transfer

Benedikt and Segoufin obtained the following decidable characterisation of FO-definable tree languages.

Theorem 4.4.1 ([BS09a]). *A tree language is* FO*-definable iff it is regular, aperiodic, and closed under guarded swaps.*

The constituent properties of this characterisation are defined as follows.

Definition 4.4.2 (Aperiodicity). A tree language L is *aperiodic* if there exists a constant $\ell \in \mathbb{N}$ such that

$$C^{\ell} \cong_L C^{\ell+1}$$

for all contexts C.

Up to minor notational differences, the following definitions of the swap operations and the claimed facts about them are from [BS09a].

Definition 4.4.3 (Horizontal Swap). Let t be a tree containing nodes u, v with $u \parallel v$. Consider the decomposition of t into a 2-context $C := t[\epsilon, u, v)$ and trees $s_1 := t_{|u}$, and $s_2 := t_{|v}$, i.e.

$$t = C[s_1, s_2].$$

The *horizontal swap of* t *between* u *and* v is the tree

$$\mathrm{hs}(t, u, v) \ := \ C[s_2, s_1].$$

If u and v are k-similar, then $\mathrm{hs}(t, u, v)$ is k*-guarded*.

Definition 4.4.4 (Vertical swap). Let t be a tree which contains nodes $u_1 \trianglelefteq v_1 \trianglelefteq u_2 \trianglelefteq v_2$. Consider the decomposition of t into contexts $C := t[\epsilon, u_1)$, $C_1 := t[u_1, v_1)$, $D := t[v_1, u_2)$, $C_2 := t[u_2, v_2)$, and a tree $s := t_{|v_2}$, i.e.

$$t = C \cdot C_1 \cdot D \cdot C_2 \cdot s.$$

The *vertical swap* of t between u_1, v_1 and u_2, v_2 is the tree

$$\mathrm{vs}(t, u_1, v_1, u_2, v_2) \ := \ C \cdot C_2 \cdot D \cdot C_1 \cdot s.$$

The vertical swap is k*-guarded* if u_1, u_2 are k-similar and v_1, v_2 are k-similar.

A tree t' is a *swap of* t if it is either a horizontal or a vertical swap of t. See Figure 4.4.1 for an illustration of both kinds of swaps. If we speak about $\mathrm{hs}(t, u, v)$ or $\mathrm{vs}(t, u_1, v_1, u_2, v_2)$ without introducing the corresponding nodes beforehand, we assume that the nodes satisfy the conditions for a swap. Note that a k-guarded swap is also k'-guarded, for each $k' \leqslant k$.

If t' is a swap of t, each node x of t corresponds to a unique node $\pi(x)$ of t' and if t' is k-guarded, then $\pi(x)$ has the same $(k+1)$-type in t' as x in t. For instance, if $t' = \mathrm{hs}(t, u, v)$ and $x = uy$, then $\pi(x) = vy$ which clearly has the same $(k+1)$-type as x. The complete definition of the bijection π is straightforward, if somewhat cumbersome in the case of vertical swaps, and will not be necessary. But note that its existence implies that $t =_{k+1} t'$ for each k-guarded swap t' of a tree t.

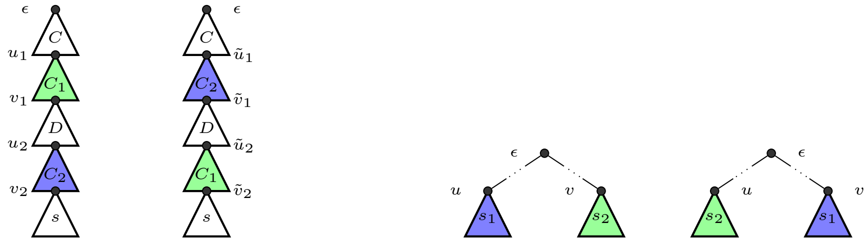


Figure 4.4.1: A tree t and its vertical swap $\mathrm{vs}(t, u_1, v_1, u_2, v_2)$ (left) and another tree t and its horizontal swap $\mathrm{hs}(t, u, v)$ (right). The nodes $\tilde{u}_1, \tilde{v}_1, \tilde{u}_2, \tilde{v}_2$ are the nodes which correspond to the nodes u_1, v_1, u_2, v_2 of t in $\mathrm{vs}(t, u_1, v_1, u_2, v_2)$.

A tree language L is *closed under guarded swaps* if there is a k such that for each tree t and each of its k-guarded vertical swaps t', if $t \in L$ then also $t' \in L$. The following characterisation of the $\mathrm{FO}_{\mathrm{card}}$-definable *word languages* was obtained by Segoufin and Schweikardt.

Theorem 4.4.5 ([SS10])**.** *A word language L is $\mathrm{FO}_{\mathrm{card}}$-definable iff it is regular and closed under guarded swaps[1] and under transfer.*

Here, a regular word language L is *closed under transfer* iff for all words x, y, z with $|x| = |z|$,

$$x^{\omega+1} y z^{\omega} \cong_L x^{\omega} y z^{\omega+1}.$$

Our goal for this section is a generalisation of Theorem 4.4.5 to regular tree languages. As a first step, we generalise the transfer operation to tree languages. Similarly to guarded swaps, we need a “vertical” and a “horizontal” operation.

Definition 4.4.6 (Transfer)**.** A regular tree language L is *closed under vertical transfer* if $C_1^{\omega+1} \cdot D \cdot C_2^{\omega} \cong_L C_1^{\omega} \cdot D \cdot C_2^{\omega+1}$ for all contexts C_1, D, C_2 with $|C_1| = |C_2|$. L is *closed under horizontal transfer* if the trees $C[C_1^{\omega+1} \cdot s_1,\, C_2^{\omega} \cdot s_2]$ and $C[C_1^{\omega} \cdot s_1,\, C_2^{\omega+1} \cdot s_2]$ agree on L, for all 2-contexts C, contexts C_1, C_2 with $|C_1| = |C_2|$, and trees s_1 and s_2. A language L is *closed under transfer* if it is both closed under horizontal and closed under vertical transfer.

Now that we have introduced the necessary notation, we can state our first main theorem.

Theorem 4.4.7 ($\mathrm{FO}_{\mathrm{card}}$-definable tree languages)**.** *A tree language L is $\mathrm{FO}_{\mathrm{card}}$-definable iff it is regular, closed under guarded swaps, and closed under transfer.*

Before we come to the proof of the characterisation, we need to introduce some further concepts. The definition of closure under transfer makes the connection between the

[1]Note that horizontal swaps are meaningless for words. The characterisation of [SS10] is stated in terms of a property of the syntactic monoid of L which, for regular word languages, is equivalent to closure under guarded swaps (cf. [BS09a, Theorem 6] and our discussion in Section 4.6.2).

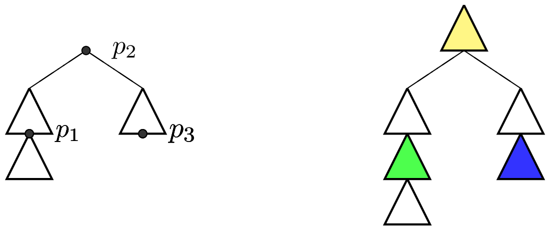


Figure 4.4.2: A 3-template T (left) and the tree $T\langle C_1, C_2, C_3\rangle$ (right) obtained by insertion of the contexts $C_1 :=$▲, $C_2 :=$▲, and $C_3 :=$▲ at p_1, p_2, p_3, respectively.

transfer operation for word languages and the one for tree languages apparent. In particular, the vertical transfer is a straightforward translation of the transfer operation to trees. We rephrase the transfer property in terms of *templates* which we introduce below. A template is a tree with several distinguished nodes which we call *points*. In contrast to the holes of a context, points are not necessarily leaves. While we operate on contexts by substitution, for templates we allow *insertion* of contexts between a point and its predecessor (if it exists). Before we proceed with the formal definition, see Figure 4.4.2 for an example. Rephrasing the transfer property in terms of templates highlights the fact that the relative position of the contexts in the surrounding tree is actually unimportant. This makes it more convenient for the proof of Theorem 4.4.7.

Definition 4.4.8 (Insertion)**.** Consider a tree t and a context C and let $p \in \mathrm{dom}(t)$. Then $t = t[\epsilon, p] \cdot t_{|p}$. The *insertion of* C at p in t is defined as the tree

$$t\langle p \hookleftarrow C\rangle \;:=\; t[\epsilon, p] \cdot C \cdot t_{|p}.$$

This definition is inductively extended to several pairwise distinct nodes $p_1, \ldots, p_n \in \mathrm{dom}(t)$ and several contexts $C_1, \ldots, C_n$ as follows. Let p'_n be the node which corresponds to p_n in $t\langle p_1 \hookleftarrow C_1, \ldots, p_{n-1} \hookleftarrow C_{n-1}\rangle$. Then

$$t\langle p_1 \hookleftarrow C_1, \ldots, p_n \hookleftarrow C_n\rangle \;:=\; (t\langle p_1 \hookleftarrow C_1, \ldots, p_{n-1} \hookleftarrow C_{n-1}\rangle)\langle p'_n \hookleftarrow C_n\rangle$$

Definition 4.4.9 (Template)**.** An *n-template* $T := (t, p_1, \ldots, p_n)$ is a tree t with $n \geqslant 0$ pairwise distinct nodes $p_1, \ldots, p_n$ of t. The nodes $p_1, \ldots, p_n$ are the *points* of T and t is its *underlying tree*. The insertion operation is extended to templates by defining

$$T\langle C_1, \ldots, C_n\rangle \;:=\; t\langle p_1 \hookleftarrow C_1, \ldots, p_n \hookleftarrow C_n\rangle.$$

We also define an insertion operation where only the points with indices belonging to a set $I := \{i_1, \ldots, i_j\} \subseteq [1, n]$ are used and the remaining points are left unchanged. To this end, we define

$$T\langle C_1, \ldots, C_j\rangle_I \;:=\; (t\langle p_{i_1} \hookleftarrow C_1, \ldots, p_{i_j} \hookleftarrow C_j\rangle, p'_1, \ldots, p'_{n-j}),$$

where p'_i is the point which corresponds in $t\langle p_{i_1} \hookleftarrow C_1, \ldots, p_{i_j} \hookleftarrow C_j\rangle$ to the i-th point in the sequence $(p_k)_{k\in[1,n]\setminus I}$.

The following straightforward lemma restates the definition of transfer in terms of templates.

Lemma 4.4.10 (Transfer for templates)**.** *A regular tree language L is* closed under transfer *iff for all 2-templates T and all contexts C_1, C_2 with $|C_1| = |C_2|$ the trees $T\langle C_1^{\omega+1}, C_2^{\omega}\rangle$ and $T\langle C_1^{\omega}, C_2^{\omega+1}\rangle$ are L-equivalent.*

Proof. Let C_1, C_2 be contexts such that $|C_1| = |C_2|$. We first show the "only if"-direction. Let T be a 2-template with points p_1, p_2. Let $t := T\langle C_1^{\omega+1}, C_2^{\omega}\rangle$ and $t' := T\langle C_1^{\omega}, C_2^{\omega+1}\rangle$.
Consider the case that $p_1 \lhd p_2$. Let $T = \Delta_1 \cdot \Delta_2 \cdot s$ for $\Delta_1 := T[\epsilon, p_1)$, let $\Delta_2 := T[p_1, p_2)$, and let $s := T_{|p_2}$. By Definition 4.4.9, we have $t = \Delta_1 \cdot C_1^{\omega+1} \cdot \Delta_2 \cdot C_2^{\omega} \cdot s$. By the closure of L under vertical transfer, $C_1^{\omega+1} \cdot \Delta_2 \cdot C_2^{\omega} \cong_L C_1^{\omega} \cdot \Delta_2 \cdot C_2^{\omega+1}$. Hence, t is L-equivalent to $t' = \Delta_1 \cdot C_1^{\omega} \cdot \Delta_2 \cdot C_2^{\omega+1} \cdot s$.
The case $p_2 \lhd p_1$ can be treated analogously.
Consider the case where $p_1 \parallel p_2$. Let $C := T[\epsilon, p_1, p_2)$ and $s_1 := T_{|p_1}$, $s_2 := T_{|p_2}$, so that $T = C[s_1, s_2]$. By Definition 4.4.9, we have $t = C[C_1^{\omega+1} \cdot s_1, C_2^{\omega} \cdot s_2]$. By closure of L under horizontal transfer, t is L-equivalent to $t' = C[C_1^{\omega} \cdot s_1, C_2^{\omega+1} \cdot s_2]$.

Now we turn to the "if"-direction. To show that L is closed under horizontal transfer, let C be a 2-context with holes h_1, h_2 and let s_1 and s_2 be trees. Let T be the template $(C[s_1, s_2], h_1, h_2)$. This way $C[C_1^{\omega+1} \cdot s_1, C_2^{\omega} \cdot s_2] = T\langle C_1^{\omega+1}, C_2^{\omega}\rangle$, which, by assumption, is L-equivalent to $C[C_1^{\omega} \cdot s_1, C_2^{\omega+1} \cdot s_2] = T\langle C_1^{\omega}, C_2^{\omega+1}\rangle$. The right-hand trees in these equations are L-equivalent to our assumption. Hence, L is closed under horizontal transfer
To show that L is closed under vertical transfer, let C, D be contexts, and let t be a tree. Let h_1 be the hole of C and h_2 the hole of $C \cdot D$. Let T be the template $(C \cdot D \cdot t, h_1, h_2)$. This way, $C \cdot C_1^{\omega+1} \cdot D \cdot C_2^{\omega} \cdot t = T\langle C_1^{\omega+1}, C_2^{\omega}\rangle$ and $C \cdot C_1^{\omega} \cdot D \cdot C_2^{\omega+1} \cdot t = T\langle C_1^{\omega}, C_2^{\omega+1}\rangle$. The right-hand trees in these equations are L-equivalent according to our assumption. Hence, L is closed under vertical transfer. $\square$

4.4.2 From definability to closure properties

We come to the proof of Theorem 4.4.7. As usual in such results, the implication from closure properties to definability is the easy part and the reverse implication is much harder. We begin with the former part. That is, we prove the following lemma.

Lemma 4.4.11. *Each $\mathrm{FO}_{\mathrm{card}}$-definable tree language is regular, closed under guarded swaps, and closed under transfer.*

First, we establish a simple normal form for $\mathrm{FO}_{\mathrm{card}}$-formulae.

Lemma 4.4.12 (Normal form for $\mathrm{FO}_{\mathrm{card}}$-formulae)**.**
For each $\mathrm{FO}_{\mathrm{card}}$-formula φ there exists a number $m \in \mathbb{N}^+$ and FO-formulae $\varphi_0, \ldots, \varphi_{m-1}$, such that

$$\varphi \equiv \bigvee_{a \in [m]} \big(\varphi_a \wedge \mathrm{C}_{a,m}\big). \tag{4.1}$$

Furthermore, each of the formulae $\varphi_0, \ldots, \varphi_{m-1}$ has exactly the same free variables as φ.

Proof. The proof proceeds by induction on the structure of the formula φ.

For the induction base, let φ be an atomic $\mathrm{FO}_{\mathrm{card}}$-formula. If φ is either $x = y$, $\mathrm{S}_i(x, y)$, or $\mathrm{P}_c(x)$, choose $m := 1$ and $\varphi_0 := \varphi$. As all trees t satisfy $|t| \equiv 0 \pmod 1$, (4.1) holds for this choice of m and φ_0. If $\varphi = C_{a,m}$, choose m according to the formula. For all $b \in [m]$, let

$$\varphi_b \;:=\; \begin{cases} \forall x\, x = x & \text{, if } b = a, \\ \exists x\, x \neq x & \text{otherwise.} \end{cases}$$

In the induction step, we distinguish several cases depending on the form of φ. Suppose that $\varphi = \exists x\ \psi$. By induction there exist a number m_ψ and FO-formulae ψ_a, for all $a \in [m]$, such that:

$$\varphi \;\equiv\; \exists x \bigvee_{a\in[m_\psi]} (\psi_a \wedge C_{a,m}) \;\equiv\; \bigvee_{a\in[m_\psi]} \big((\exists x\ \psi_a) \wedge C_{a,m}\big).$$

Set $m := m_\psi$ and $\varphi_a := \exists x\ \psi_a$, for all $a \in [m]$. The invariant regarding the free variables is obviously satisfied.

Suppose now that $\varphi = \psi_1 \wedge \psi_2$. By induction, we obtain $m_1, m_2 \in \mathbb{N}$ and FO-formulae $\psi_{1,a}$ and $\psi_{2,b}$, for $a \in [m_1]$, $b \in [m_2]$, such that

$$\psi_1 \;\equiv\; \bigvee_{a\in[m_1]} (\psi_{1,a} \wedge C_{a,m_1}) \qquad \text{and} \qquad \psi_2 \;\equiv\; \bigvee_{a\in[m_2]} (\psi_{2,a} \wedge C_{a,m_2}).$$

It is easily seen that for $i \in [1, 2]$ and for all $k \in \mathbb{N}^+$,

$$\psi_i \equiv \bigvee_{a\in[km_i]} (\psi_{i,(a \,\mathrm{rem}\, m_i)} \wedge C_{a,km_i}).$$

Now, for $m := m_1 \cdot m_2$, obviously

$$\begin{aligned} \varphi \;&\equiv\; \bigvee_{a\in[m]} (\psi_{1,(a \,\mathrm{rem}\, m_1)} \wedge C_{a,m}) \;\wedge\; \bigvee_{b\in[m]} (\psi_{2,(b \,\mathrm{rem}\, m_2)} \wedge C_{b,m}) \\ &\equiv \bigvee_{(a,b)\in[m]\times[m]} \big(\psi_{1,(a \,\mathrm{rem}\, m_1)} \wedge \psi_{2,(b \,\mathrm{rem}\, m_2)} \wedge C_{a,m} \wedge C_{b,m}\big). \end{aligned}$$

As $C_{a,m} \wedge C_{b,m}$ cannot be satisfied for distinct numbers a and b, this implies for $\varphi_a := \psi_{1,(a \,\mathrm{rem}\, m_1)} \wedge \psi_{2,(a \,\mathrm{rem}\, m_2)}$, for all $a \in [m]$, that

$$\varphi \;\equiv\; \bigvee_{a\in[m]} \big(\varphi_a \wedge C_{a,m}\big).$$

Each of the formulae φ_a obviously contains the same free variables as φ.

Next, consider the case that $\varphi = \neg\psi$. By the induction hypothesis, ψ is equivalent to a formula of the form of (4.1) for some modulus m and formulae ψ_a, for all $a \in [m]$. That is,

$$\psi \;\equiv\; \bigvee_{a\in[m]} \big(\psi_a \wedge C_{a,m}\big).$$

Note that for each tree t and each suitable variable assignment β, (t,β) satisfies ψ iff (t,β) satisfies ψ_r for $r := |t| \operatorname{rem} m$. Thus, (t,β) satisfies $\neg\psi$ iff (t,β) satisfies $\neg\psi_r$. Therefore,

$$\varphi \;=\; \neg\psi \;\equiv\; \bigvee_{a\in[m]} \big(\neg\psi_a \wedge C_{a,m}\big) \;=\; \bigvee_{a\in[m]} \big(\varphi_a \wedge C_{a,m}\big),$$

where $\varphi_a := \neg\psi_a$, for all $a \in [m]$. Furthermore, each of the formulae φ_a obviously contains the same free variables as φ. □

For each $m \in \mathbb{N}$, let $T_{i,m}$ denote the set of trees of size i modulo m.

Lemma 4.4.13. *Let L be a tree language such that*

$$L \;=\; \bigcup_{a\in[m]} (L_a \cap T_{a,m}),$$

for some $m \in \mathbb{N}^+$ and tree languages $L_0, \ldots, L_{m-1}$.

1. *If each of the languages $L_0, \ldots, L_{m-1}$ is aperiodic, then L is closed under transfer.*
2. *If each of the languages $L_0, \ldots, L_{m-1}$ is closed under guarded swaps, then L is closed under guarded swaps.*

Proof.

1. Suppose that the languages $L_0, \ldots, L_{m-1}$ are all aperiodic. Let T be a 2-template and let C_1, C_2 be contexts such that $|C_1| = |C_2|$, and consider the trees $t := T\langle C_1^{\omega+1}, C_2^{\omega}\rangle$ and $t' := T\langle C_1^{\omega}, C_2^{\omega+1}\rangle$. Let $a := |t| \operatorname{rem} m$ and observe that $a = |t'| \operatorname{rem} m$. We have $t \in L$ iff $t \in L_a \cap T_{a,m}$ iff, applying the aperiodicity of L_a twice, $t' \in L_a \cap T_{a,m}$ iff $t' \in L$.
2. Suppose that the languages $L_0, \ldots, L_{m-1}$ are all closed under guarded swaps. For each $a \in [m]$, let k_a be such that L_a is closed under k_a-guarded swaps. Let $k := \max_{a\in[m]} k_a$. Observe that L_a is also closed under k-guarded swaps. Note that for each tree t and each k-guarded swap t' of t, we have $|t| = |t'|$. Hence, $L_a \cap T_{a,m}$ is also closed under k-guarded swaps. Altogether, we obtain that L is closed under k-guarded swaps.

□

With these preparations in place, the proof of Lemma 4.4.11 follows immediately.

Proof of Lemma 4.4.11. Let L be a tree language that is $\mathrm{FO}_{\mathrm{card}}$-definable by a sentence φ. By Lemma 4.4.12, we have

$$\varphi \;\equiv\; \bigvee_{a\in[m]} \big(\varphi_a \wedge \mathrm{C}_{a,m}\big),$$

for some number $m \in \mathbb{N}^+$ and FO-sentences $\varphi_0, \ldots, \varphi_{m-1}$. According to Theorem 4.4.1, for each $a \in [m]$, the language L_a defined by φ_a is regular, aperiodic and closed under swaps. Since the class of regular tree languages is closed under intersection and union, L is also regular. Applying Lemma 4.4.11, we obtain the desired result. □

4.4.3 From closure properties to definability

Theorem 4.4.7 will be obtained from the following lemma. Recall that we write $s \equiv_q t$ if the trees s and t satisfy the same FO-sentences of quantifier-rank at most q. If, furthermore, $|s| \equiv |t| \pmod m$, we write $s \equiv_q^m t$.

Lemma 4.4.14 (Main Lemma)**.** *Let L be a regular tree language. If L is closed under guarded swaps and closed under transfer, then there exist $m, q \in \mathbb{N}$ such that L is a union of $\equiv_q^m$-equivalence classes.*

The Main Lemma immediately implies Theorem 4.4.7. It shows that L is a union of $\equiv_q^m$-equivalence classes. There are only finitely many classes and each such class can be defined by an $\mathrm{FO}_{\mathrm{card}}$-sentence. The disjunction of these sentences defines L.

The outline of the proof of the Main Lemma follows the proof of Theorem 4.4.1 in [BS09a]. For the proof, we need some definitions. When we obtain a context $C = t[u, v)$ from a tree t, the types of nodes which are close to the hole of C are not necessarily the same as the types of the corresponding nodes in t. As a remedy, we add the information about the k-type of v in t to the context. To this end, we define $t[u, v)_k := (t[u, v), t^k_{|v})$. A pair (C, τ) such that $(C, \tau) = t[u, v)_k$ for some tree t and nodes $u, v \in \mathrm{dom}(t)$ is a *k-abstract context*. We let $|(C, \tau)| := |C|$. Consider some $k' \leqslant k + 1$. We define the k'-type of a node x of C in (C, τ) as the k'-type of x in the tree $C \cdot \tau$. Observe that the k'-type of a node x in (C, τ) is the same as the k'-type of ux in any tree t such that $(C, \tau) = t[u, v)_k$. Our notation for speaking about k'-types in trees extends to k-abstract contexts. If a tree s or a k-abstract context (C', τ') is k-similar to τ, we say that it is *compatible* with (C, τ). In this case, we define the concatenation $(C, \tau) \cdot (C', \tau') := (C \cdot C', \tau')$, which is again a k-abstract context, and we let $(C, \tau) \cdot s := C \cdot s$. If (C, τ) is compatible with itself, it is a *k-abstract loop*. Using these definitions, we can extend definitions based on context concatenation (e.g. L-equivalence, template insertion) to k-abstract contexts. Note that our definitions are slightly different but equivalent to the definitions of [BS09a].

We start with the proof of the Main Lemma where we introduce several lemmas which will be proved afterwards.

Proof of Lemma 4.4.14. Let L be a regular tree language that is closed under transfer and under k-guarded swaps, for some $k \in \mathbb{N}$. We show that there are $q \in \mathbb{N}$ and $m \in \mathbb{N}^+$ such that for all trees s and t, if $s \equiv_q^m t$ then $s \in L$ iff $t \in L$.

Our first major step (Lemma 4.4.15 below) can be summarised very roughly as follows. We show that q can be chosen such that if s and t satisfy the same FO-sentences of quantifier-rank at most q and have the same size modulo m, then either (1) s and t have the same number of occurrences of each $(k + 1)$-type and this implies that s and t are L-equivalent (in this case we are done), or (2) there is a tree t' which is L-equivalent to t such that each $(k + 1)$-type τ is contained either as often in t' as in s or τ occurs strictly more often in t'. Furthermore, In case (2) we get a set of k-abstract loops which contain all types which occur more often in t' then in s.

Using regularity of L, we can replace all abstract loops by L-equivalent abstract loops whose size is bounded in terms of L and which have the same size modulo m (Lemma 4.4.16).

Then we show that we can simultaneously embed copies of some of these loops into a tree which is L-equivalent to s and which contains each $(k+1)$-type as often as s (Lemma 4.4.17). Up to this point, we will have used only regularity and closure under k-guarded swaps. We show that for regular tree languages closure under transfer is equivalent to closure under an operation which we call m-transfer (cf. Lemma 4.4.19; this fixes the value of m). Using this operation, we can add a copy of each loop and hence of the types contained in the loops without affecting membership in L. Repeating this argument, we end up with a tree s' which is L-equivalent to s and which contains each $(k+1)$-type exactly as often as t'. By (1) above, this shows that s' and t' are L-equivalent, and hence s and t are L-equivalent and we are done.

The first step is done almost exactly as in the proof of Benedikt and Segoufin's characterisation of FO-definability [BS09a, Theorem 2] and is summarised by the following lemma. Only minor modifications are necessary to adapt everything to our notation and to care for the size of the considered trees modulo m.

Lemma 4.4.15 (Type Equivalence)**.** *Let L be a regular tree language that is closed under k-guarded swaps, for some $k \in \mathbb{N}$.*

1. *If s, t are trees such that $s =_{k+1} t$ and s, t are $(k+1)$-similar, then s, t are L-equivalent.*
2. *For all $d, m \in \mathbb{N}$ there exists a $q \in \mathbb{N}$ such that for all trees s, t with $s \equiv_q^m t$ and not $s =_{k+1} t$, there exists a tree t' and a sequence of k-abstract loops $(S_i)_{i\in[1,n]}$, for some $n \in \mathbb{N}^+$, such that*
   a) *t' is $(k+1)$-similar to s,*
   b) *$|t'| \equiv |s| \pmod m$,*
   c) *the trees t' and t are L-equivalent,*
   d) *for each $(k+1)$-type τ,*
      i. *$|t'|_\tau = |s|_\tau + |(S_i)_{i\in[1,n]}|_\tau$,*
      ii. *$|s|_\tau > d$ if τ occurs in S_i.*

We fix q according to Lemma 4.4.15(2) for some value of d which we choose below. If $s =_{k+1} t$, we are done by Lemma 4.4.15(1). Assume that $s =_{k+1} t$ does not hold and let $(S_i)_{i\in[1,n]}$ and t' be given as in Lemma 4.4.15(2). Note that Lemma 4.4.15(2d) implies, in particular, that $s =_{k+1} t'$ does not hold. We will construct a new tree s' which is $(k+1)$-similar and L-equivalent to s and which contains all $(k+1)$-types which occur more often in t' than in s. Precisely, we want to achieve $|s'|_\tau = |s|_\tau + |(S_i)_{i\in[1,n]}|_\tau$ for all $(k+1)$-types τ. That is, $s' =_{k+1} t'$ and s' is $(k+1)$-similar to t'. From Lemma 4.4.15(1) we know that t' and s' are L-equivalent. Since t' and t as well as s' and s are L-equivalent, we know that s and t are L-equivalent and we are done with the proof.

As a first step in the construction of s', we replace the loops $S_1, \dots, S_n$ by equivalent loops of small size, but with the same size modulo m.

Lemma 4.4.16 (Context Bound). *Let $k \in \mathbb{N}$, $m \in \mathbb{N}^+$, and let L be a regular tree language. There exists a computable bound $b := b(L, m, k) \in \mathbb{N}$ such that for all k-abstract loops C there exists a k-abstract loop C' satisfying:*

1. $C' \cong_L C$,
2. $|C'| \leqslant b$,
3. $|C'| \equiv |C| \pmod{m}$,
4. $C' \leqslant_{k+1} C$.

Let b be given by the lemma and let $S'_i = C'$ be the loop given by the lemma for $C = S_i$, for each $i \in [1, n]$. Since $|t'|_\tau = |s|_\tau + |(S_i)_{i\in[1,n]}|_\tau$, for each type τ, in particular $|t'| = |s| + \sum_{i\in[1,n]} |S_i|$. Hence, $\sum_{i\in[1,n]} |S_i| \equiv 0 \pmod{m}$, since $|t'| \equiv |s| \pmod{m}$. Since also $\sum_{i\in[1,n]} |S'_i| \equiv 0 \pmod{m}$, a simple application of the pigeon hole principle yields a non-empty set $I := \{i_1, \dots, i_\ell\} \subseteq [1, n]$, for some $\ell \leqslant m$, such that $|S'_{i_1}| + \cdots + |S'_{i_\ell}| \equiv 0 \pmod{m}$. The next lemma shows that we can turn s into an L-equivalent tree which contains disjoint copies of the loops $(S'^{\omega}_i)_{i\in I}$.

Lemma 4.4.17 (Disjoint Context Inclusion). *Let L be a regular tree language which is closed under k-guarded swaps, for some $k \in \mathbb{N}$. Let $(C_i)_{i\in[1,\ell]}$, for $\ell \in \mathbb{N}^+$, be a sequence of k-abstract loops. For all trees s such that $(C_i)_{i\in[1,\ell]} <_{k+1} s$, there exists an ℓ-template T such that $T\langle C_1, \dots, C_\ell\rangle$ is L-equivalent to s, $T\langle C_1, \dots, C_\ell\rangle =_{k+1} s$, and $T\langle C_1, \dots, C_\ell\rangle$ is $(k + 1)$-similar to s.*

For the application of the lemma to $(S'^{\omega}_i)_{i\in I}$, we have to be sure that $(S'_i)_{i\in I} <_{k+1} s$. This can be achieved by taking $d := m\omega b$, since each type which occurs in one of the loops occurs more than d times in s according to Lemma 4.4.15(2). Obviously, there cannot be more occurrences of any particular $(k + 1)$-type in $(S'_i)_{i\in I}$ than there are nodes in $(S'_i)_{i\in I}$ altogether. Let T be given by Lemma 4.4.17 for s and $(C_i)_{i\in[1,\ell]} = (S'_i)_{i\in I}$.

Our next step is to use the closure of L under transfer to add copies of the loops $(S'_i)_{i\in I}$. For this, we show that closure under transfer is equivalent to the following property.

Definition 4.4.18 (m-transfer). A regular tree language L is closed under *m-transfer* if for all $\ell \in \mathbb{N}^+$, all contexts $C_1, \dots, C_\ell$, all ℓ-templates T and all $\delta_1, \dots, \delta_\ell \in \mathbb{N}$, if $\delta_1|C_1| + \delta_2|C_2| + \cdots + \delta_\ell|C_\ell| \equiv 0 \pmod{m}$, then $T\langle C_1^\omega, \dots, C_\ell^\omega\rangle$ and $T\langle C_1^{\omega+\delta_1}, \dots, C_\ell^{\omega+\delta_\ell}\rangle$ are L-equivalent.

Lemma 4.4.19 (Modulo Transfer). *A regular tree language L is closed under transfer iff L is closed under m-transfer for $m := \mathrm{lcm}[1, \kappa_L]$.*

The previous lemma fixes the value of m. By Lemma 4.4.19, we know that $T\langle S'^{\omega}_{i_1}, \dots, S'^{\omega}_{i_\ell}\rangle$ is L-equivalent to $T\langle S'^{\omega+1}_{i_1}, \dots, S'^{\omega+1}_{i_\ell}\rangle$ which is L-equivalent to $T\langle S^{\omega+1}_{i_1}, \dots, S^{\omega+1}_{i_\ell}\rangle$ according to Lemma 4.4.16. Note that we have added exactly one copy of each context from $(S_i)_{i\in I}$ to a tree which is L-equivalent to s. In particular, the number of occurrences of each $(k + 1)$-type τ in the new tree is $|s|_\tau + |(S'_i)_{i\in I}|_\tau$.

We iterate this argument to add copies of the remaining loops to obtain the desired tree s' such that $|s'|_\tau = |s|_\tau + |(S_i)_{i\in[1,n]}|_\tau$ for each $(k + 1)$-type τ. This finishes the proof of Lemma 4.4.14. □

Proof of the Modulo Transfer Lemma

The Modulo Transfer Lemma (Lemma 4.4.19) adapts a lemma which was proved for regular word languages in [SS10]. The proof follows [SS10] closely, but some details have to be adapted since the role that was formerly played by the *length* of words is now played by the *size* of a tree. In particular, this changes the way how we obtain the value of m for which we show that L is closed under m-transfer. For this, we show that every regular tree language which is closed under transfer has the following property.

Definition 4.4.20 (Quasi-aperiodicity)**.** A regular tree language L is *quasi-aperiodic with parameter* m if $C^\omega \cong_L C^{\omega+1}$, for all contexts C with $|C| \equiv 0 \pmod{m}$. A regular language is *quasi-aperiodic* if it is quasi-aperiodic with parameter m, for some $m \in \mathbb{N}^+$.

The notion of quasi-aperiodicity has been studied before for languages of words, see e.g. [Str94].

Lemma 4.4.21. *A regular tree language L which is closed under transfer is also quasi-aperiodic with parameter $m := \operatorname{lcm}[1, \kappa_L]$.*

Proof. Let C be a context with $|C| \equiv 0 \pmod{m}$. We have to show that $C^\omega \cong_L C^{\omega+1}$.

If $|C| = 0$, then C is the empty context, and $C^\omega \cong_L C^{\omega+1}$ is satisfied trivially.

If $|C| \geqslant 1$, we proceed as follows. Recall that, according to our definition of κ, there is a context E with $1 \leqslant |E| \leqslant \kappa$ and $E \cong_L C^\omega$. By our definition of m, there exists a $j \in \mathbb{N}^+$ such that $|E^j| = |C|$. Since C^ω is idempotent, the same is true for E. Hence, $E^j \cong_L E \cong_L C^\omega$. Let $D := E^j$. As $|D| = |C|$, by the closure of L under transfer, $C^{\omega+1}D^\omega \cong_L C^\omega D^{\omega+1}$. For the left side, we have $C^{\omega+1}D^\omega \cong_L C^{\omega+1}(C^\omega)^\omega \cong_L C^{\omega+1}$, and for the right side we have $C^\omega D^{\omega+1} \cong_L C^\omega (C^\omega)^{\omega+1} \cong_L C^\omega$. Hence, $C^\omega \cong_L C^{\omega+1}$. $\square$

Proof of Lemma 4.4.19. For proving the implication from m-transfer to transfer, let C_1, C_2 be contexts with $|C_1| = |C_2|$, and let T be a 2-template. We want to show that $t := T\langle C_1^{\omega+1}, C_2^\omega\rangle$ and $t' := T\langle C_1^\omega, C_2^{\omega+1}\rangle$ are L-equivalent. Let $T = (s, p_1, p_2)$ and define the 2-template $T' := (s\langle p_2 \hookleftarrow C_2\rangle, p_1', p_2')$ where p_1' and p_2' correspond to p_1 and p_2 in $s\langle p_2 \hookleftarrow C_2\rangle$. We have $T'\langle D_1, D_2\rangle = T\langle D_1, C_2 \cdot D_2\rangle$, for all contexts D_1, D_2. Let $\delta_1 := \omega m + 1$ and $\delta_2 := \omega m - 1$, so that $\delta_1|C_1| + \delta_2|C_2| = 2\omega m|C_1| \equiv 0 \pmod{m}$. By assumption, we know that $T'\langle C_1^{\omega+\delta_1}, C_2^{\omega+\delta_2}\rangle$ and $T'\langle C_1^\omega, C_2^\omega\rangle = T\langle C_1^\omega, C_2 \cdot C_2^\omega\rangle = t'$ are L-equivalent. But $T'\langle C_1^{\omega+\delta_1}, C_2^{\omega+\delta_2}\rangle = T'\langle C_1^{\omega+\omega m+1}, C_2^{\omega+\omega m-1}\rangle = T\langle C_1^{\omega+\omega m+1}, C_2^{\omega+\omega m}\rangle \cong_L T\langle C_1^{\omega+1}, C_2^\omega\rangle = t$. Hence, L is closed under transfer.

For the reverse direction in the statement of the lemma, we let $m \in \mathbb{N}^+$ be given according to Lemma 4.4.21 so that L is quasi-aperiodic for the constant m. As an immediate consequence of the quasi-aperiodicity of L, we know that for all contexts C and all $j \in \mathbb{N}$:

$$j|C| \equiv 0 \pmod{m} \implies C^\omega \cong_L C^{\omega+j}. \tag{4.2}$$

The proof proceeds by induction on the number ℓ of points of T. Let $T = (s, p)$ be a 1-template. Let C_1 be a context and $\delta_1 \in \mathbb{N}$, such that $\delta_1|C_1| \equiv 0 \pmod{m}$. Let $C := T[\epsilon, p)$

and $t := T_{|p}$. By Definition 4.4.9, $T\langle C_1^\omega\rangle = C \cdot C_1^\omega \cdot t$. The right-hand tree is L-equivalent to $C \cdot C_1^{\omega+\delta_1} \cdot t = T\langle C_1^{\omega+\delta_1}\rangle$ by (4.2).

Consider now the case that $\ell > 1$. For all $i \in [1,\ell]$, let $\alpha_i := |C_i|$. Define $d := \gcd\{m, \alpha_2, \ldots, \alpha_\ell\}$. By our initial assumption about the size of the contexts modulo m, the number d also divides $\delta_1\alpha_1$. Let $m' := m/d$ and $\alpha_i' := \alpha_i/d$, for all $i \in [1,\ell]$.

Notice that $|C_1^{\delta_1\alpha_j'}| = |C_j^{\delta_1\alpha_1'}|$, for all $j \in [1,\ell]$. Using this, we may apply the closure of L under transfer to replace copies of C_1 by copies of C_j in any 2-template T' to obtain

$$T'\langle C_1^{\omega\delta_1\alpha_j'+\delta_1\alpha_j'}, C_j^{\omega\delta_1\alpha_1'}\rangle \in L \iff T'\langle C_1^{\omega\delta_1\alpha_j'}, C_j^{\omega\delta_1\alpha_1'+\delta_1\alpha_1'}\rangle \in L.$$

As $C^{a\omega} \cong_L C^\omega$, for all contexts C and $a \in \mathbb{N}^+$, we have

$$T'\langle C_1^{\omega+\delta_1\alpha_j'}, C_j^{\omega}\rangle \in L \iff T'\langle C_1^{\omega}, C_j^{\omega+\delta_1\alpha_1'}\rangle \in L, \tag{4.3}$$

By repeatedly applying (4.3), we obtain for all $i \in \mathbb{N}^+$ that

$$T'\langle C_1^{\omega+i\delta_1\alpha_j'}, C_j^{\omega}\rangle \in L \iff T'\langle C_1^{\omega}, C_j^{\omega+i\delta_1\alpha_1'}\rangle \in L, \tag{4.4}$$

By Bézout's identity (a well-known fact of elementary number theory, see e.g. [JJ98]) there exist $r_1, \ldots, r_k \in \mathbb{Z}$ such that

$$d = \gcd\{m, \alpha_2, \ldots, \alpha_\ell\} = r_1 m + r_2\alpha_2 + \cdots + r_\ell\alpha_\ell.$$

Dividing by d and multiplying by δ_1, we have

$$\delta_1 = \delta_1 r_1 m' + \delta_1 r_2 \alpha_2' + \delta_1 r_\ell \alpha_\ell'.$$

Let $t := T\langle C_1^{\omega+\delta_1}, \ldots, C_\ell^{\omega+\delta_\ell}\rangle$. Our aim is to change $C_1^{\omega+\delta_1}$ into C_1^ω which then allows us to apply the induction hypothesis to finish the proof. By the last equation, we have

$$t \;=\; T\langle C_1^{\omega+\delta_1 r_1 m'+\delta_1 r_2\alpha_2'+\cdots+\delta_1 r_\ell\alpha_\ell'}, C_2^{\omega+\delta_2}, \ldots, C_\ell^{\omega+\delta_\ell}\rangle.$$

We want to use Equation (4.2) to delete some repetitions of C_1 and Equation (4.4) to change the remaining repetitions into copies of the other contexts. To this end, it is necessary to replace each negative r_i by a positive number r_i' such that $r_i' \equiv r_i \pmod m$. That is, $r_i' = r_i + p_i m$ for some $p_i \in \mathbb{N}^+$. Let $P := \delta_1 m(p_1 m' + p_2\alpha_2' + \cdots + p_\ell\alpha_\ell')$. Since, trivially, $Pm|C_1| \equiv 0 \pmod m$, we can apply (4.2) to add Pm repetitions of C_1 in the previous decomposition of t to obtain that t is L-equivalent to

$$T\langle C_1^{\omega+\delta_1 r_1' m'+\delta_1 r_2'\alpha_2'+\cdots+\delta_1 r_\ell'\alpha_\ell'}, C_2^{\omega+\delta_2}, \ldots, C_\ell^{\omega+\delta_\ell}\rangle.$$

Recall that $\alpha_1 = \alpha_1' d$ and $m' = m/d$ and hence $\alpha_1 d$ is a multiple of m. Thus $r_1'\delta_1 m'\alpha_1 \equiv 0 \pmod m$ which means that we can apply equation (4.2) to get rid of $r_1'\delta_1 m'$ copies of C_1 in this tree. That is, t is L-equivalent to

$$T\langle C_1^{\omega+\delta_1 r_2'\alpha_2'+\cdots+\delta_1 r_\ell'\alpha_\ell'}, C_2^{\omega+\delta_2}, \ldots, C_\ell^{\omega+\delta_\ell}\rangle.$$

Now, by applying (4.4) successively between the first and the i-th point of T for each $i \in [2, \ell]$, we can replace $r'_i \delta_1 \alpha'_i$ copies of C_1 at the first point of T by $r'_i \delta_1 \alpha'_1$ copies of C_i at the i-th point. We obtain that t is L-equivalent to

$$T\langle C_1^\omega, C_2^{\omega+\delta_2+r'_2\delta_1\alpha'_1}, \ldots, C_\ell^{\omega+\delta_\ell+r'_\ell\delta_1\alpha'_1}\rangle.$$

For all $i \in [1, \ell]$, let $\delta'_i := \delta_i + r'_i \delta_1 \alpha'_1$. Let $T' := T\langle C_1^\omega \rangle_{\{1\}}$. Then T' is an $(\ell - 1)$-template and

$$T'\langle C_2^{\omega+\delta'_2}, \ldots, C_\ell^{\omega+\delta'_\ell}\rangle = T\langle C_1^\omega, C_2^{\omega+\delta'_2}, \ldots, C_\ell^{\omega+\delta'_\ell}\rangle.$$

Let

$$\delta := \delta'_2\alpha_2 + \cdots + \delta'_\ell\alpha_\ell \;=\; \delta_2\alpha_2 + \cdots + \delta_\ell\alpha_\ell + \delta_1\alpha'_1(r'_2\alpha_2 + \cdots + r'_\ell\alpha_\ell).$$

From the definition of d and $r'_1, \ldots, r'_k$ above, we have $r'_2\alpha_2 + \cdots + r'_\ell\alpha_\ell \equiv d \,(\mathrm{mod}\, m)$, and hence $\delta \equiv \delta_1\alpha_1 + \cdots + \delta_\ell\alpha_\ell \equiv 0 \,(\mathrm{mod}\, m)$, since $\delta_1\alpha'_1 d = \delta_1\alpha_1$ by definition of α'_1. This observation allows to apply the induction hypothesis to the $(\ell - 1)$-template T' and the contexts $C_2, \ldots, C_\ell$:

$$T'\langle C_2^{\omega+\delta'_2}, \ldots, C_\ell^{\omega+\delta'_\ell}\rangle \in L \iff T'\langle C_2^\omega, \ldots, C_\ell^\omega\rangle \;=\; T\langle C_1^\omega, C_2^\omega, \ldots, C_\ell^\omega\rangle \in L$$

In summary: $T\langle C_1^{\omega+\delta_1}, C_2^{\omega+\delta_2}, \ldots, C_\ell^{\omega+\delta_\ell}\rangle \in L$ iff $T\langle C_1^\omega, C_2^\omega, \ldots, C_\ell^\omega\rangle \in L$. □

Proof of the Disjoint Context Inclusion Lemma

In this section, we prove Lemma 4.4.17. We say that a node v in a tree t is *compatible* with a k-abstract context C, if $t_{|v}$ is compatible with C. The proof of Lemma 4.4.17 uses the following lemma.

Lemma 4.4.22. *Let $k \in \mathbb{N}$. Let $C_1, \ldots, C_n$ be k-abstract loops. Let t be a tree containing nodes $p_1, \ldots, p_n$ and $p'_1, \ldots, p'_n$ such that both p_i and p'_i are compatible with C_i, for all $i \in [1, n]$. Let $T := (t, p_1, \ldots, p_n)$ and $T' := (t, p'_1, \ldots, p'_n)$. Then the tree $T'\langle C_1, \ldots, C_n\rangle$ can be obtained from the tree $T\langle C_1, \ldots, C_n\rangle$ by a sequence of k-guarded swaps.*

Proof. It suffices to consider the case of $n = 1$ and the general claim follows by induction. That is, we show that for each k-abstract-context C, and all nodes p, p' of t which are compatible with C, the tree $t\langle p \hookleftarrow C\rangle$ can be turned into the tree $t\langle p' \hookleftarrow C\rangle$ by a sequence of k-guarded swaps.

By definition, we have $t\langle p \hookleftarrow C\rangle = t[\epsilon, p)Ct_{|p}$ and $t\langle p' \hookleftarrow C\rangle = t[\epsilon, p')Ct_{|p'}$. Let h be the hole of C. As p and C as well as p' and C are compatible, the nodes h, p, p', ph and $p'h$ are pairwise k-similar. We distinguish three cases depending on the relative positions of p and p', see the figures 4.4.3 and 4.4.4.

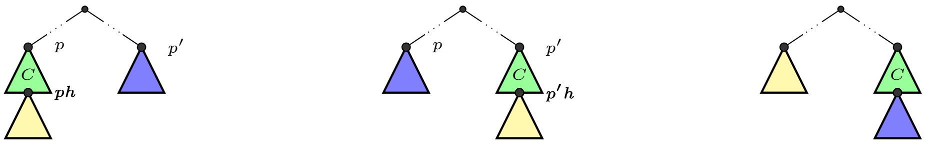


Figure 4.4.3: Case 1 in the proof of Lemma 4.4.22. The trees from left to right are $t\langle p \hookleftarrow C\rangle$, $s = \mathrm{hs}(t\langle p \hookleftarrow C\rangle, p, p')$, and $\mathrm{hs}(s, p, p'h) = t\langle p' \hookleftarrow C\rangle$.

Case I: $p' \parallel p$.
We have $t\langle p \hookleftarrow C\rangle = t[\epsilon, p)Ct_{|p} = t[\epsilon, p, p')[Ct_{|p}, t_{|p'}]$. Let $s := \mathrm{hs}(t\langle p \hookleftarrow C\rangle, p, p')$ which is a k-guarded swap. Then $s = t[\epsilon, p, p')[t_{|p'}, Ct_{|p}]$ and $\mathrm{hs}(s, p, p'h) = t[\epsilon, p, p')[t_{|p}, Ct_{|p'}] = t[\epsilon, p')Ct_{|p'}$ is a k-guarded swap. See Figure 4.4.3 for an illustration.

Case II: $p \lhd p'$.
We have $t\langle p \hookleftarrow C\rangle = t[\epsilon, p)Ct[p, p')t_{|p'}$. Then $\mathrm{vs}(t\langle p \hookleftarrow C\rangle, p, ph, p', p') = t[\epsilon, p)t[p, p')Ct_{|p'}$ is a k-guarded swap and we have $t[\epsilon, p)t[p, p')Ct_{|p'} = t\langle p' \hookleftarrow C\rangle$. See Figure 4.4.4 for an illustration.

Figure 4.4.4: Case 2 in the proof of Lemma 4.4.22. Left is the tree $t\langle p \hookleftarrow C\rangle$ and right is $\mathrm{vs}(t\langle p \hookleftarrow C\rangle, p, ph, p', p') = t\langle p' \hookleftarrow C\rangle$.

.

Case III: $p' \lhd p$.
Analogously to Case 2 with the roles of p and p' reversed.

□

For the following proof of Lemma 4.4.17, we need the notion of a k-*inclusion* from [BS09a]:

Definition 4.4.23 (k-Inclusion)**.** Let s, t be trees. A mapping $f : \mathrm{dom}(s) \to \mathrm{dom}(t)$ is a k-*inclusion*, if

1. f *preserves* k-*types*, i.e. $s^k_{|v} = t^k_{|f(v)}$, for all $v \in \mathrm{dom}(s)$,
2. f *preserves the successor relations*, i.e. if $u = vi$, then $f(u) = f(v)i$, for all $u, v \in \mathrm{dom}(s)$ and $i \in \Pi$.

Note that this definition extends naturally to k-abstract contexts.

The Disjoint Context Inclusion Lemma is an extension of the following lemma to an arbitrary number of loops.

Lemma 4.4.24 (Lemma 4, [BS09b]). *Let $k \in \mathbb{N}^+$. Let C be a k-abstract loop, and let s be a tree such that $C <_{k+1} s$. There exists a tree s' obtained from s by a sequence of k-guarded swaps, such that (1) C is $(k+1)$-included in s', (2) $s' =_{k+1} s$, and (3) $|s'| = |s|$.*

Note that the statement about the size of s and s' follows immediately from the construction of s' which uses only swaps.

We are now ready for the proof of the Disjoint Context Inclusion Lemma.

Proof of Lemma 4.4.17. Instead of $(k+1)$-inclusions, we will simply speak of inclusions. We include the loops $(C_i)_{i\in[\ell+1]}$ one after another in s. To this end, we construct a sequence $(T_i)_{i\in[\ell+1]}$ of templates with $T_0 := s$ where for all $i \in [1,\ell]$ the following conditions are satisfied:

1. $T_i := (s_i, p_{i,1}, \dots, p_{i,i})$ is an i-template,
2. $p_{i,j}$ is compatible with C_j, for all $j \in [1,i]$,
3. $T_i\langle (C_j)_{j\in[1,i]}\rangle =_{k+1} s$,
4. $T_i\langle (C_j)_{j\in[1,i]}\rangle$ and s are L-equivalent, and
5. $T_i\langle (C_j)_{j\in[1,i]}\rangle$ is $(k+1)$-similar to t.

The tree $T_0 = s$ is a 0-template and $s\langle\rangle = s$. We assume that, for some $i \in \mathbb{N}^+$, we have constructed a sequence of trees satisfying the conditions. Let $C := C_{i+1}$. Because each C_j is a loop and $p_{i,j}$ is compatible with C_j in s_i, we have $|s_i|_\tau = |s|_\tau - |(C_j)_{j\in[1,i]}|_\tau$, for each $(k+1)$-type τ. By the initial assumption about the types in the lemma, $|s|_\tau - |(C_j)_{j\in[1,i]}|_\tau > \big|(C_j)_{j\in[1,\ell]}\big|_\tau - \big|(C_j)_{j\in[1,i]}\big|_\tau \geqslant |C|_\tau$. Thus $C <_{k+1} s_i$ and hence, by Lemma 4.4.24, we obtain an inclusion e of C in a tree s_i' obtained from s_i by k-guarded swaps. Let $p_{i,1}', \dots, p_{i,i}'$ be the nodes of s_i' which correspond to the points of T_i. From the definition of guarded swaps, we know that each $p_{i,j}'$ is $(k+1)$-similar in s_i' to the node $p_{i,j}$ in s_i. Hence, $p_{i,j}$ is compatible with C_j. Let C' denote the image of the domain of C under the inclusion e. Lemma 4.4.24 does not tell us anything about the position of C' in s_i' and so it might be the case that C' contains one or more of the points $p_{i,1}', \dots, p_{i,i}'$. Consider some $p_{i,j}'$ which belongs to C'. Since $C <_{k+1} s_i$ and $s_i =_{k+1} s_i'$, by Lemma 4.4.24 we know that the number of occurrences of the $(k+1)$-type τ outside C' is at least as large as the number of occurrences inside C'. Hence, there is a node $p_{i,j}''$ which does not belong to C' and which is compatible with C_j. For each $p_{i,j}'$ which does not belong to C', we let $p_{i,j}'' := p_{i,j}'$. Let $T_i' := (s_i', p_{i,1}'', \dots, p_{i,i}'')$. By Lemma 4.4.22, $T_i'\langle C_1, \dots, C_i\rangle$ can be obtained from $T_i\langle C_1, \dots, C_i\rangle$ by a sequence of k-guarded swaps. Thus, $T_i'\langle C_1, \dots, C_i\rangle$ is L-equivalent to $T_i\langle C_1, \dots, C_i\rangle$ and hence also to s. Furthermore, $T_i'\langle C_1, \dots, C_i\rangle =_{k+1} T_i\langle C_1, \dots, C_i\rangle =_{k+1} s$, and $T_i'\langle C_1, \dots, C_i\rangle$ is $(k+1)$-similar to $T_i\langle C_1, \dots, C_i\rangle$. Since e is an inclusion, we obtain that $s_i' = s_i'[\epsilon, e(\epsilon)) \cdot C \cdot s'_{i|e(\epsilon)h}$, where h is the hole of C. To obtain T_{i+1}, we let $s_{i+1} := s_i'[\epsilon, e(\epsilon)) \cdot s'_{i|e(\epsilon)h}$ and $p_{i+1,i+1} := e(\epsilon)$ and we let $p_{i+1,1}, \dots, p_{i+1,i}$ be the nodes of s_{i+1} which correspond to the nodes $p''_{i,1}, \dots, p''_{i,i}$ of s'_i. It is easy to see that the induction hypothesis is preserved by this construction. $\square$

Proof of the Context Bound Lemma

The Context Bound Lemma (Lemma 4.4.16) is proved by a standard pumping argument. For the proof, we need the following definitions and the following (standard) observation. Recall that for a minimal deterministic tree automaton $\mathcal{A}$ with state set Q, for each state $p \in Q$ we can choose a tree t_p such that the run of $\mathcal{A}$ on t_p reaches the state p at the root, and hence each context C induces a *transition function* on Q which maps each state $p \in Q$ to the state $q \in Q$ reached by the run of $\mathcal{A}$ on Ct_p at the root.

Definition 4.4.25 ($\mathcal{A}$-labelling)**.** Let $\mathcal{A}$ be a minimal deterministic tree automaton with state set Q. Let C be a context with the hole h. The *$\mathcal{A}$-labelling* of C labels each node v of C by a function from Q to Q as follows:

- if $v \trianglelefteq h$, then the label of v is the transition function induced by $C_{|v}$ on Q.
- if $v \parallel h$, then v is labelled by the constant function taking each state to the state reached by $\mathcal{A}$ at the root of $C_{|v}$.

Observation 4.4.26. *Let $\mathcal{A}$ be a minimal deterministic tree automaton with state set Q. Let C be a context. Let $x \triangleleft y$ be nodes of C such that (1) x and y receive the same label by the $\mathcal{A}$-labelling of C, and (2) for the hole h of C, either $x \triangleleft y \trianglelefteq h$ or $x \parallel h$. Let C' be the context obtained from C by removing the context $C[x, y)$ from C. Then $C \cong_L C'$.*

(Note that $C' = C[\epsilon, x)t_{|y}$ if $x \triangleleft y \trianglelefteq h$ and $C' = C[\epsilon, x, h)[x \leftarrow t_{|y}]$ if $x \parallel h$.)

Proof. To see that this is true, observe that, if $x \triangleleft y \trianglelefteq h$, then the transition function induced by the context is the identity function on Q, and thus $C[x, y)$ can safely be removed from C. If otherwise x and h are incomparable, the state reached by $\mathcal{A}$ on the subtree containing x is not changed by removing $C[x, y)$. $\square$

Now we can prove the Context Bound Lemma.

Proof of Lemma 4.4.16. Let $k \in \mathbb{N}, m \in \mathbb{N}^+$ and let $\mathcal{A}$ be a minimal deterministic tree automaton with state set Q which recognises a tree language L. We show that each k-abstract loop C whose height is strictly larger than $\nu := 2mK|Q^Q|$, where K denotes the number of $(k+1)$-types, is L-equivalent to a strictly smaller k-abstract loop C' such that $|C'| \equiv |C| \,(\mathrm{mod}\, m)$ and $C' \leqslant_{k+1} C$.

This proves the lemma since we consider trees of rank r and hence each tree of height at most ν has size at most $r^{\nu+1}$, which yields the bound b of the lemma.

Now consider a k-abstract loop C with a hole h and suppose that C contains a leaf z such that $|z| \geqslant \nu + 1$. Let γ denote the $\mathcal{A}$-labelling of C and let us label each node x above z by

$$(\gamma(x), C^k_{|x}, |C_{|x}| \,\mathrm{rem}\, m).$$

Observe that there are at most $\nu/2$ different labels of this kind. Hence, there exist nodes x_1, x_2, x_3 with $x_1 \lhd x_2 \lhd x_3 \unlhd z$ which all receive the same label. Note that, for $i = 1$ or $i = 2$, we must have $x_i \lhd x_{i+1} \unlhd h$ or $x_i \parallel h$; for if not $x_2 \parallel h$, then $x_1 \lhd x_2 \unlhd h$, and if not $x_2 \lhd x_3 \unlhd h$ then either $x_2 \parallel h$ or $x_1 \lhd x_2 \unlhd h$. Hence, and because $\gamma(x_1) = \gamma(x_2) = \gamma(x_3)$, the conditions of Observation 4.4.26 are satisfied for $x := x_i$ and $y := x_{i+1}$. This means we can remove the context $C[x, y)$ to obtain a strictly smaller context C' which is L-equivalent to C.

Since $C^k_{|x} = C^k_{|y}$, the context $(C[x, y), C^k_{|y})$ is a k-abstract loop. Removing it from C does not introduce any new $(k + 1)$-types. That is, all nodes of C' have the same $(k + 1)$-types as the corresponding nodes of C. Hence, $C' \leqslant_{k+1} C$. In particular, the $(k + 1)$-type of the root and the hole of C' are not changed, and hence C' is also a k-abstract loop.

Note that $|C_{|y}| = |C_{|x}| - |C[x, y)|$. Since $|C_{|x}| \operatorname{rem} m = |C_{|y}| \operatorname{rem} m$, this implies $|C[x, y)| \equiv 0 \pmod{m}$. Hence, $|C'| = |C| - |C[x, y)| \equiv |C| \pmod{m}$. □

Proof of the Type Equivalence Lemma

The proof of the Type Equivalence Lemma (Lemma 4.4.15) is implicitly contained in the proof of [BS09a, Theorem 2]. For the sake of completeness, we show how the precise statement can be obtained from the proof of [BS09a, Theorem 2]. To this end, we need to introduce some notation from [BS09a]. We write $s =_{k,d} t$ to express that $|s|_\tau, |t|_\tau > d$ holds for each k-type τ such that $|s|_\tau \neq |t|_\tau$. We use $s \leqslant_{k,d} t$ as a shorthand for $s =_{k,d} t$ and $s \leqslant_k t$.

The first step towards the proof of Lemma 4.4.15 is the following adaptation of Lemma 1 of [BS09a]:

Lemma 4.4.27. *Let $k', m, d' \in \mathbb{N}$. There exists a $q \in \mathbb{N}$ such that for all trees s, t, if $s \equiv^m_q t$, then the following holds:*

1. $s =_{k',d'} t$,
2. $|s| \equiv |t| \pmod{m}$,
3. *s and t are k'-similar.*

Proof. Let $\tau_1, \ldots, \tau_\nu$ be a list of all trees of depth at most k'. The (k', d', m)-*type* of a tree s is the tuple $(n_1, \ldots, n_\nu, a, \tau)$ such that

- $n_i = \min\{n'_i, d' + 1\}$, where $n'_i := |t|_{\tau_i}$ is the number of nodes in s of k'-type τ_i, for each $i \in [1, \nu]$,
- $a = |s| \operatorname{rem} m$, and
- τ is the k'-type of the root of s.

Note that two trees s and t satisfy the lemma's conditions (1)–(3) iff they have the same (k', d', m)-type. Furthermore, the number of (k', d', m)-types is finite. Thus, it suffices to construct, for each (k', d', m)-type α, a $\mathrm{FO}_{\mathrm{card}}$-sentence φ_α that is satisfied by exactly the trees of (k', d', m)-type α. The number q mentioned in the lemma is then chosen as the maximum quantifier depth of the sentences φ_α, for all (k', d', m)-types α.

For a (k', d', m)-type α of the form $(n_1, \ldots, n_\nu, a, \tau)$, the formula φ_α can be chosen as

$$\varphi_\alpha \ := \bigwedge_{i\in[1,\nu]} \varphi_{i,n_i} \ \wedge \ \mathrm{C}_{a,m} \ \wedge \ \varphi_\tau,$$

where

- for every $i \in [1, \nu]$,
  - if $n_i \leqslant d'$, then φ_{i,n_i} states that there are exactly n_i nodes whose k'-type is τ_i,
  - if $n_i = d' + 1$, then φ_{i,n_i} states that there are at least $d' + 1$ nodes whose k'-type is τ_i,
- $\mathrm{C}_{a,m}$ states that the size of the tree is congruent to a modulo m, and
- φ_τ states that τ is the k'-type of the root of the tree.

The formulae φ_{i,n_i} and φ_τ can easily be constructed, once we have available formulae $\psi^{k'}_{t'}(x)$, for each tree t' of depth at most k', expressing that the k'-type of node x is t'.

The construction of the formulae $\psi^{k'}_{t'}(x)$ is straightforward: We inductively construct for all numbers $k \leqslant k'$ and for all nodes v of t' a formula $\psi^k_{t',v}(x)$, such that for all trees s and all nodes x of s, $s \models \psi^k_{t',v}(x)$ iff the k-type of x in s equals the k-type of v in t'. Then, $\psi^{k'}_{t'}(x)$ can be chosen as $\psi^{k'}_{t'}(x) := \psi^{k'}_{t',w}(x)$, where w is the root of t'. We define $\psi^k_{t',v}(x)$ for all nodes v of t' carrying the label $a \in \Sigma$ as follows:

$$\begin{aligned}
\psi^0_{t',v}(x) \ &:= \mathrm{P}_a(x),\\
\psi^k_{t',v}(x) \ &:= \mathrm{P}_a(x) \ \wedge \ \varphi^k_{t',v,1}(x) \ \wedge \ \varphi^k_{t',v,2}(x),
\end{aligned}$$

where $\varphi^k_{t',v,i}(x)$ describes the i-successor c_i of v, i.e. for all $i \in \Pi$:

$$\begin{aligned}
\varphi^k_{t',v,i}(x) &:= \exists y \ \big(\mathrm{S}_i(x,y) \wedge \psi^{k-1}_{t',c_i}(y)\big) \quad , \text{ if } vi \in \mathrm{dom}(t),\\
\varphi^k_{t',v,i}(x) &:= \forall y \ \neg \mathrm{S}_i(x,y) \quad , \text{ otherwise.}
\end{aligned}$$

□

The next lemma is a slight strengthening of [BS09a, Lemma 2]:

Lemma 4.4.28 ([BS09a, Lemma 2]). *Let L be a regular tree language that is closed under k-guarded swaps. Let $m, d \in \mathbb{N}$. There exists a $d' \in \mathbb{N}$ such that, for all trees s, t, if $s =_{k+1,d'} t$, then there exists a tree t' with the following properties:*

1. $s \leqslant_{k+1,d} t'$,
2. *t' and t are $(k+1)$-similar,*

3. *t' and t are L-equivalent,*

4. *$|t'| \equiv |t| \, (mod \, m)$.*

Proof. The only difference to [BS09a, Lemma 2] is property 4 of t'. The proof in [BS09a] uses a pumping argument. It is easy to see that the pumping in the construction of the tree t' from t can be done without changing the size of t modulo m. □

We need the following definition of [BS09a]:

Definition 4.4.29 (pseudo-inclusion)**.** A tree t is *k-pseudo-included* in a tree t' if there is an injective mapping $h : \mathrm{dom}(t) \to \mathrm{dom}(t')$ such that:

1. $h(\epsilon) = \epsilon$,

2. h *preserves k-types* , i.e. $t^k_{|v} = t'^k_{|h(v)}$, for all $v \in \mathrm{dom}(t)$,

3. h maps the i-successor of each node x to a node in the subtree of the image of x , i.e. if $x = yi$ then $h(x) \trianglerighteq h(y)i$, for all $x, y \in \mathrm{dom}(t)$ and $i \in \Pi$.

The following lemma and its corollary are Lemma 3 and Corollary 1 of [BS09a], respectively.

Lemma 4.4.30 (Lemma 3, [BS09a])**.** *Let L be a regular tree language that is closed under k-guarded swaps. Let s, t' be $(k+1)$-similar trees such that $s \leqslant_{k+1} t'$. There exists a tree t'' such that:*

1. *s is $(k+1)$-pseudo-included in t'',*

2. *t'' and t' are L-equivalent,*

3. *t'' and t' are $(k+1)$-similar,*

4. *$t'' =_{k+1} t'$,*

5. *$|t''| = |t'|$.*

The statement about the size of the tree t'' in Lemma 4.4.30 is not formulated in [BS09a], but it is justified by the fact that the construction of the tree t'' from the tree t' in the proof of Lemma 3 uses swapping operations only, which do not change the size of a tree.

Corollary 4.4.31 (Corollary 1, [BS09a])**.** *Let L be a regular tree language that is closed under k-guarded swaps. If s and t are $(k+1)$-similar trees with $s =_{k+1} t$, then s and t are L-equivalent.*

We are now ready for the proof of the Type Equivalence Lemma.

Proof of Lemma 4.4.15. Part (a) holds by Corollary 4.4.31.

For the proof of part (b), choose d' according to Lemma 4.4.28, and choose q according to Lemma 4.4.27 for $k' := k + 1$. Thus, if $s \equiv_q^m t$, then we know that $s =_{k+1,d'} t$, s and t are $(k + 1)$-similar, and $|s| \equiv |t| \pmod m$. By Lemma 4.4.28, we know that there exists a tree t' satisfying the requirements of Lemma 4.4.30. Hence, we obtain a tree t'' with the properties stated therein. In particular, s is $(k + 1)$-pseudo-included in t'' by a map h.

Since not $s =_{k+1} t'$ and h is a pseudo-inclusion, we obtain that there exist nodes x and y such that $x = yi$ and $h(x) \rhd h(y)i$, for some $i \in \{0, 1\}$. We let $(x_1, y_1), \ldots, (x_n, y_n)$ denote all pairs of such nodes. Then we define $S_j := t''[h(y_j)i, h(x_j))_k$ for each such pair (x_j, y_j) and the corresponding i. Since h is a $(k + 1)$-inclusion, we know that the $(k + 1)$-type of $h(y_j)i$ and of $h(x_j)$ is the same. Hence, each S_j is a k-abstract loop.

Now consider some $(k + 1)$-type τ. Since h is a $(k + 1)$-inclusion, τ occurs as often in the h-image of s in t'' as in s. Hence, $|t'|_\tau = |t''|_\tau = |s|_\tau + |(S_i)_{i\in[1,n]}|_\tau$. Furthermore, if τ occurs in one of the loops S_j, then $|s|_\tau \neq |t''|_\tau$. Since $s \leqslant_{k+1,d} t'$ and $t' =_{k+1} t''$, we have $|s|_\tau > d$. □

4.5 Decidability of $\mathrm{FO_{card}}$-definability

In this section we discuss how our characterisation of $\mathrm{FO_{card}}$-definability leads to decidability. That is, we show:

Theorem 4.5.1. *There is an algorithm which on input of a tree automaton $\mathcal{A}$ decides if the tree language L accepted by $\mathcal{A}$ is* $\mathrm{FO_{card}}$*-definable.*

By Theorem 4.4.7, we know that it suffices to check if L is closed under guarded swaps and closed under transfer. The following theorem was proved in [BS09a].

Theorem 4.5.2. *There is a polynomial time algorithm which on input of a deterministic tree automaton $\mathcal{A}$ decides if the tree language L accepted by $\mathcal{A}$ is closed under guarded swaps.*

Hence, to establish Theorem 4.5.1, it suffices to prove the following lemma.

Lemma 4.5.3. *There is an algorithm which on input of a tree automaton $\mathcal{A}$ decides if the tree language L accepted by $\mathcal{A}$ is closed under transfer*

For the proof of Lemma 4.5.3, we first prove a more amenable characterisation of closure under transfer. For this, we will need the following pumping lemma for templates.

Lemma 4.5.4 (Template Bound). *Let L be a regular tree language and let $\ell \in \mathbb{N}$. There exists a number $B := B(L, \ell)$ (which can be computed from a tree automaton recognising L) such that for each ℓ-template T there exists an L-equivalent ℓ-template T' of size at most B. That is, T' is a template whose underlying tree has at most B nodes such that for all contexts $C_1, \ldots, C_\ell$, we have*

$$T\langle C_1, \ldots, C_\ell\rangle \in L \iff T'\langle C_1, \ldots, C_\ell\rangle \in L.$$

Proof. Let $m(T)$ denote the maximum number of points on any path of T. We proceed by induction on $m(T)$.

If $m(T) = 0$, then T is just a tree and the statement of the lemma reduces to the statement of the standard pumping lemma for regular tree languages.

Suppose that $T = (t, p_1, \ldots, p_\ell)$ is a template with $m(T) > 0$. We consider the set M of minimal points of T, i.e. M is an antichain in the tree order (i.e. if $p, p' \in M$ and $p \neq p'$, then $p \parallel p'$) and the elements of M are the minima of their respective chains (i.e. if $p \in M$ and $p' \notin M$, then $p \lhd p'$).

Let $M = \{m_1, \ldots, m_k\}$. For each m_i, we let $s_i := t_{|m_i}$ and we let $\bar{v}_i$ denote a sequence which contains all points p of T such that $m_i \lhd p$, in arbitrary order. For each s_i, we construct the template $S_i := (s_i, \bar{v}_i)$ which has strictly less points then T since m_i is not contained in $\bar{v}_i$.

We consider the k-context $C := t[\epsilon, m_1, \ldots, m_k)$. We have $t = C[s_1, \ldots, s_k]$. Using a standard pumping lemma for contexts, we obtain a context C' which is L-equivalent to C and whose size is bounded by a constant b which depends only on L and the number of holes which is at most ℓ. By construction of the templates S_i, we obviously have $m(S_i) < m(T)$ for each S_i. We apply the induction hypothesis to obtain a template S'_i of size at most $B(L, \ell - 1)$ for each template S_i. For each S'_i, we let s'_i denote its underlying tree. We define $t' := C'[s'_1, \ldots, s'_k]$ and we let T' be the ℓ-template with the underlying tree t' whose sequence of points is $m_1, \ldots, m_k$ followed by the concatenation of the points of the templates $S'_1, \ldots, S'_k$, where each point of S'_i is prefixed by m_i. Note that the size if T' is at most $B(L, \ell) := b + \ell B(L, \ell - 1)$.

Now consider contexts $C_1, \ldots, C_\ell$. It is straightforward to verify (cf. Definition 4.4.8) that the insertion of contexts at parallel points is equivalent to context substitution in such a way that $T\langle m_1 \hookleftarrow C_1, \ldots, m_k \hookleftarrow C_k\rangle = C[C_1 s_1, \ldots, C_k s_k]$. For each m_i, we let $\bar{C}_i$ denote the contexts from $\{C_1, \ldots, C_\ell\}$ which, in the insertion operation $T\langle C_1, \ldots, C_\ell\rangle$, are inserted at points which lie in the subtree s_i below m_i. By the construction of the templates S_i, we hence have: $T\langle C_1, \ldots, C_\ell\rangle = C[C_1 S_1\langle \bar{C}_1\rangle, \ldots, C_k S_k\langle \bar{C}_k\rangle]$. Since C' is L-equivalent to C and each tree $S_i\langle \bar{C}_i\rangle$ is L-equivalent to the corresponding tree $S'_i\langle \bar{C}_i\rangle$ by induction, we obtain that the trees $T\langle C_1, \ldots, C_\ell\rangle$ and $C'[C_1 S'_1\langle \bar{C}_1\rangle, \ldots, C_k S'_k\langle \bar{C}_k\rangle]$ are L-equivalent, where the latter tree is $T'\langle C_1, \ldots, C_\ell\rangle$.

□

Now we can introduce our characterisation of transfer.

Definition 4.5.5 (Bounded transfer). Let L be a regular tree language. Let $m := \mathrm{lcm}[1, \kappa_L]$. Let $b := b(L, m, 1)$ be the context bound of Lemma 4.4.16 and let $B := B(L, m)$ be the template bound of Lemma 4.5.4. We say that L is *closed under bounded transfer* if for all $\ell \leqslant m$, all contexts $C_1, \ldots, C_\ell$ of size at most b, all ℓ-templates T of size at most B, and all natural numbers $\delta_1, \ldots, \delta_\ell < m\omega_L$, if $\delta_1|C_1| + \cdots + \delta_\ell|C_\ell| \equiv 0 \pmod{m}$, then $T\langle C_1^\omega, \ldots, C_\ell^\omega\rangle$ and $T\langle C_1^{\omega+\delta_1}, \ldots, C_\ell^{\omega+\delta_\ell}\rangle$ are L-equivalent.

Lemma 4.5.6. *A regular tree language L is closed under bounded transfer iff it is closed under transfer.*

Proof. First, recall that we know from Lemma 4.4.19 that closure under transfer is equivalent to closure under m-transfer for $m = \mathrm{lcm}[1, \kappa_L]$. Hence, it suffices to establish the equivalence of closure under bounded transfer and closure under m-transfer for this choice of m.

The definition of bounded transfer is derived from the definition of m-transfer by imposing upper bounds on the various parameters. The implication from closure under m-transfer to closure under bounded transfer is hence immediate. It remains to show that imposing those upper bounds can be done without loss of generality, i.e. that closure under bounded transfer also implies closure under m-transfer.

According to Lemma 4.5.4, we can assume that the template T in the definition of m-transfer has size at most B.

Using Lemma 4.4.16, we can also assume that each of the contexts C_i in the definition of m-transfer has size at most b. (Recall that Lemma 4.4.16 tells us — if we ignore the statement about the types in that lemma — that for each C_i there is an C'_i such that $C'_i \cong_L C$, $|C'_i| \leqslant b$, $|C'_i| \equiv |C_i| \pmod{m}$; in particular, the summed size of these contexts is also 0 modulo m.)

Next, we bound the size of the numbers δ_i. Consider contexts $C_1, \ldots, C_\ell$, an ℓ-template T, and let $\delta_1, \ldots, \delta_\ell$ be such that $\delta_1|C_1| + \delta_2|C_2| + \cdots + \delta_\ell|C_\ell| \equiv 0 \pmod{m}$. Consider the tree $T\langle C_1^\omega, \ldots, C_\ell^\omega\rangle$. For each of the numbers δ_i, we choose $a_i \in \mathbb{N}$ and $r_i \in [\omega m]$ such that

$$\delta_i = a_i\omega m + r_i.$$

We have $r_1|C_1| + r_2|C_2| + \cdots + r_\ell|C_\ell| \equiv 0 \pmod{m}$ where each $r_i < \omega m$. Hence, using closure under bounded transfer, we obtain that the tree

$$T\langle C_1^\omega, \ldots, C_\ell^\omega\rangle$$

is L-equivalent to the tree

$$T\langle C_1^{\omega+r_i}, \ldots, C_\ell^{\omega+r_i}\rangle$$

which, in turn, is L-equivalent to

$$T\langle C_1^{\omega+\delta_i}, \ldots, C_\ell^{\omega+\delta_i}\rangle$$

— since for each C_i, using that the ω-power of a context is idempotent,

$$C_i^{\omega+r_i} \;\cong_L\; C_i^{\omega+a_i m\omega+r_i} \;=\; C_i^{\omega+\delta_i}.$$

It remains to justify the bound on the number ℓ of contexts. To this end, we proceed by induction on the number ℓ. If $\ell \leqslant m$, given everything that we have established so far, the definition of m-transfer and bounded transfer agree. Now suppose that $\ell > m$ and consider a tree $T\langle C_1^\omega, \ldots, C_\ell^\omega\rangle$ for contexts $C_1, \ldots, C_\ell$, an ℓ-template T, and $\delta_1, \ldots, \delta_\ell$ such that $\delta_1|C_1| + \cdots + \delta_\ell|C_\ell| \equiv 0 \pmod{m}$. Since there are more contexts than there are remainders modulo m, there must be $i_1 < i_2 \leqslant \ell$ such that $\sum_{j\in[1,i_1]} \delta_j|C_j| \equiv \sum_{j\in[1,i_2]} \delta_j|C_j| \pmod{m}$. Let $I := [i_1+1, i_2]$. Hence, $\sum_{j\in I} \delta_j|C_j| \equiv 0 \pmod{m}$ and $\sum_{j\notin I} \delta_j|C_j| \equiv 0 \pmod{m}$. Now we can apply the induction hypothesis first to the template obtained by restricting T to the points in I and afterwards to those outside of I. By induction, we obtain that the tree $T\langle(C_i^\omega)_{j\in I}\rangle_I$ is L-equivalent to $t' := T\langle(C_i^{\omega+\delta_i})_{j\in I}\rangle_I$. Let T' be the template with the underlying tree t' and whose points are the points of T whose indices belong to $\bar{I} := [1,\ell] \setminus I$. Applying the induction hypothesis once more, we obtain that $T'\langle(C_i^\omega)_{j\in\bar{I}}\rangle$ is L-equivalent to $T'\langle(C_i^{\omega+\delta_i})_{j\in\bar{I}}\rangle$. This is the same tree as $T\langle(C_i^{\omega+\delta_i})_{i\in[1,\ell]}\rangle$ and hence we are done. □

Using the notion of bounded transfer, the proof of Lemma 4.5.3 is straightforward.

Proof of Lemma 4.5.3. By Lemma 4.5.6, it suffices to present an algorithm which, on input of a tree automation, decides if the language recognised by the automaton is closed under bounded transfer. If the language is not closed under bounded transfer, there must exist a witness for the failure of the closure property. Such a witness consists of contexts $C_1, \ldots, C_\ell$ of size at most b, an ℓ-template T of size at most B, and natural numbers $\delta_1, \ldots, \delta_\ell < m\omega$ with $\delta_1|C_1| + \cdots + \delta_\ell|C_\ell| \equiv 0 \pmod{m}$ such that either $T\langle C_1^\omega, \ldots, C_\ell^\omega\rangle \in L$ and $T\langle C_1^{\omega+\delta_1}, \ldots, C_\ell^{\omega+\delta_\ell}\rangle \notin L$ or vice versa. Here, the numbers m, ω, b, B are all computable given the automaton. Since the size of the contexts, the template, and the numbers which form the witness are all bounded, the algorithm can simply enumerate all possible combinations of contexts, templates, and natural numbers whose size does not exceed the respective upper bounds. If any witness is found, the algorithm can decide that the language is not closed under transfer. If no witness is found, we can be sure that the algorithm has considered all possible witnesses and hence it can decide that the language is closed under transfer. □

From the proof above, with a little bit more care, we could obtain an elementary — i.e. k-fold exponential, for some k — upper bound on the running time of the algorithm. But the value of k is rather large and we do not know how to improve it or show that it is necessary. Hence, we do not discuss this in greater detail.

4.6 On swaps

Trying to understand various properties of k-guarded vertical swaps, it becomes evident that k-guardedness is a too lax requirement for many purposes. There are at least two reasons for this. Consider a k-guarded vertical swap $\mathrm{vs}(t, u_1, v_1, u_2, v_2)$ of a tree t. Then

1. u_1, v_1, u_2, v_2 can be arbitrarily close together, and
2. the trees induced by the nodes at distance k above v_1 and distance k above u_2 could be different.

In this section, we introduce a notion of *strong guardedness* which rules out these cases. We then show that closure under swaps is equivalent to closure under these strongly guarded swaps and horizontal swaps (Theorem 4.6.2 below). We need this result in Section 4.7.6. There are also several results of [BS09a] and [BS09b] where this stronger guardedness condition has been implicitly assumed. Our result fills the gaps in the proofs of these results. These applications are discussed in Section 4.6.2 below.

In the following definition of strong guardedness, in addition to the k-spills of nodes in trees, we need to consider their k-spheres. Observe that, for each tree t and each of its nodes u, we have $\mathcal{N}_k^t(u) \cong (t', v)$ where t' is a tree and v is the image of u under the unique isomorphism of $\mathcal{N}_k^t(u)$ and (t', v). Note also that $\mathrm{dist}^{t_1}(u, v) = \mathrm{dist}^{t_2}(u, v)$ for all trees t_1 and t_2 with $u, v \in \mathrm{dom}(t_1) \cap \mathrm{dom}(t_2)$. Hence, we can always omit the trees in the distance notation.

Definition 4.6.1 (Strongly k-guarded)**.** We say that a vertical swap $\mathrm{vs}(t, u_1, v_1, u_2, v_2)$ is *strongly k-guarded*, if u_1 and u_2 are k-similar in t, v_1 and v_2 have isomorphic *k-spheres*, and $\mathrm{dist}(u_1, v_1) \geqslant 2k + 1$, $\mathrm{dist}(v_1, u_2) \geqslant k + 1$ and $\mathrm{dist}(u_2, v_2) \geqslant 2k + 1$.

Like for k-guarded swaps, we say that a tree language L is *closed under strongly k-guarded swaps* if for each tree t which belongs to L, each of its strongly k-guarded vertical swaps also belongs to L. We say that a language L is *closed under strongly guarded swaps* if there is a $k_0 \in \mathbb{N}$ such that L is closed under strongly k-guarded swaps for each $k \geqslant k_0$. During the following section, we define several other kinds of vertical swaps. We use analogous definitions of closure of a language for all kinds of swaps. Note that this is consistent with our previous definition of closure under guarded swaps, since a tree language which is closed under k_0-guarded swaps, for some k_0 is also closed under k-guarded swaps for each $k \geqslant k_0$. The present definition has the useful feature that, given a language which is closed under several types of swaps, we can always assume that the number k_0 witnessing the closure of L is the same for all of them by taking maxima. Our goal is a proof of the following theorem.

Theorem 4.6.2. *Let L be a tree language. Then L is closed under guarded swaps iff it is closed under strongly-guarded vertical swaps and closed under guarded horizontal swaps.*

4.6.1 Proof of Theorem 4.6.2

In the proof of Theorem 4.6.2, the implication from closure under guarded swaps to closure under strongly guarded swaps and horizontal swaps is trivial, since the former notion of swap subsumes the latter. To establish the reverse implication, we have to show that each vertical swap $\mathrm{vs}(t, u_1, v_1, u_2, v_2)$ of a tree t which is K-guarded, for some sufficiently large value of K, can be obtained from t by a sequence of k-guarded horizontal and strongly k-guarded vertical swaps. The proof proceeds by considering cases of increasing "difficulty" in a series of lemmas. Each of these lemmas establishes that a particular kind of swaps can be reduced to simpler swaps if the value of K is increased, where the simplest swaps for our purpose are strongly k-guarded vertical swaps and k-guarded horizontal swaps. Next come particular vertical swaps which we call *k-trivial*.

Definition 4.6.3 (k-trivial)**.** Let $k \in \mathbb{N}$ and let t be a tree. A vertical swap $\mathrm{vs}(t, u_1, v_1, u_2, v_2)$ is *k-trivial* if $v_1 = u_1 x$ and $v_2 = u_2 x$, for $x \in \Pi^*$, and u_1 and u_2 are $(k + |x|)$-similar.

Given nodes $u_1 \trianglelefteq v_1 \trianglelefteq u_2 \trianglelefteq v_2$ of t, we say that CC_1DC_2s is the *standard decomposition* of t if $C = t[\epsilon, u_1)$, $C_1 = t[u_1, v_1)$, $D = t[v_1, u_2)$, $C_2 = t[u_2, v_2)$, and $s = t_{|v_2}$. Recall that closure under trivial vertical swaps is defined analogously to closure under strongly guarded swaps.

Lemma 4.6.4. *If a tree language L is closed under guarded horizontal swaps, then it is also closed under trivial vertical swaps.*

Proof. Suppose that L is closed under k_0-guarded horizontal swaps, for some $k_0 \in \mathbb{N}$. Let t be a tree, let $k \geqslant k_0$, and let $t' := \mathrm{vs}(t, u_1, v_1, u_2, v_2)$ be a k-trivial vertical swap of t. We show that either $t = t'$ or t' can be obtained from t by k-guarded horizontal swaps, which implies that $t' \in L$.

Let x be given as in Definition 4.6.3 and let $\ell := |x|$. We distinguish two cases.

Let $v_{1,1}, \ldots, v_{1,n}$ be an enumeration of all nodes of t such that $v_{1,j} = u_1 x_j$ for some $x_j \in \Pi^\ell$ with $x \neq x_j$, for each $j \in [1, n]$. Let $v_{2,j} := u_2 x_j$ and note that, by $(k + \ell)$-similarity, $v_{2,j}$ is a node of t which is k-similar to $v_{1,j}$, for each $j \in [1, n]$. Let $t_{i,j} := t_{|v_{i,j}}$, for each $i \in \{1, 2\}$ and $j \in [1, n]$. Let $C_1' := t[u_1, v_{1,1}, \ldots, v_{1,n}, u_1 x)$ and $C_2' := t[u_2, v_{2,1}, \ldots, v_{2,n}, u_2 x)$. Consider the standard decomposition of t. Clearly, $C_1 = C_1'[x_1 \hookleftarrow t_{1,1}, \ldots, x_n \hookleftarrow t_{1,n}]$ and $C_2 = C_2'[x_1 \hookleftarrow t_{2,1}, \ldots, x_n \hookleftarrow t_{2,n}]$. We have $C_1' = (s_1, x_1, \ldots, x_n, x)$ and $C_2' = (s_2, x_1, \ldots, x_n, x)$ for k-similar trees s_1, s_2 with $\mathrm{height}(s_1), \mathrm{height}(s_2) \leqslant k$. Hence, $s_1 = s_2$ and $C_1' = C_2'$. We obtain a tree t'' from t by horizontally swapping $v_{1,1}$ with $v_{2,1}$, then swapping $v_{1,2}$ and $v_{2,2}$ horizontally in the resulting tree, and so on for all pairs $v_{1,j}$ and $v_{2,j}$. Then,

$$\begin{aligned} t''[u_1, u_1 x) &= C_1'[x_1 \hookleftarrow t_{2,1}, \ldots, x_{2,n} \hookleftarrow t_{2,n}] = C_2'[x_1 \hookleftarrow t_{2,1}, \ldots, x_n \hookleftarrow t_{2,n}] = C_2, \\ t''[u_2, u_2 x) &= C_2'[x_1 \hookleftarrow t_{1,1}, \ldots, x_{1,n} \hookleftarrow t_{1,n}] = C_1'[x_1 \hookleftarrow t_{1,1}, \ldots, x_n \hookleftarrow t_{1,n}] = C_1, \end{aligned}$$

and the other contexts in the decomposition of t remain unaffected by the horizontal swaps. Altogether, we obtain that

$$t'' = C C_2 D C_1 s = t'.$$

$\square$

Definition 4.6.5 (k-short)**.** Let $k \in \mathbb{N}$ and let t be a tree. A vertical swap $\mathrm{vs}(t, u_1, v_1, u_2, v_2)$ is *k-short* if it is $(k + \mathrm{dist}(u_1, v_2))$-guarded.

Lemma 4.6.6. *If a tree language L is closed under guarded horizontal swaps, then it is also closed under short vertical swaps.*

For the proof of Lemma 4.6.6, we will distinguish between the case where the path from u_1 to v_1 is the same word as the path from u_2 to v_2 and the case where those paths are different. In the latter case, we can prove a stronger statement which we will use repeatedly in the proof of Lemma 4.6.6.

Lemma 4.6.7. *Each vertical swap $\mathrm{vs}(t, u_1, v_1, u_2, v_2)$ of a tree t where u_1 and u_2 are $(k+\ell)$-similar, for some $\ell \in \mathbb{N}^+$, v_1 and v_2 are k-similar, and either*

1. *$v_1 = u_1 x$ for some $x \in \Pi^\ell$, $\mathrm{dist}(u_2, v_2) \geqslant \ell$, and $u_2 x \parallel v_2$, or*
2. *$v_2 = u_2 x$ for some $x \in \Pi^\ell$, $\mathrm{dist}(u_1, v_1) \geqslant \ell$, and $u_1 x \parallel v_1$*

can be obtained from t using only k-guarded horizontal swaps.

Proof. We consider the first case where $u_2x \parallel v_2$; the second case follows by symmetric reasoning. Since $\mathrm{dist}(u_2, v_2) \geqslant \ell$, there is an $x' \in \Pi^\ell$ such that $u_2x' \trianglelefteq v_2$. Since u_1 and u_2 are $(k+\ell)$-similar, u_1x' is a node of t which is k-similar to u_2x'. Let $C_1' := t[u_1, v_1, u_1x')$ and $s_1 := t_{|u_1x'}$. Let $C_2' := t[u_2, u_2x, u_2x')$, $s_2 := t_{|u_2x}$, and $B := t[u_2x', v_2)$. Note for later use that C_1' and C_2' are $(k+\ell)$-similar, that Bs and s_1 are k-similar, and that s_2 is k-similar to s (since s_2 is the subtree below $u_1x = v_1$ and s is the subtree below v_2, and v_1 and v_2 are k-similar in t).

Observe that $C_1 = C_1'[x' \hookleftarrow s_1]$ and $C_2 = C_2'[s_2, B]$. Hence,

$$t = C \cdot C_1'[x' \hookleftarrow s_1] \cdot D \cdot C_2'[s_2, B] \cdot s.$$

Since $C_1'[x' \hookleftarrow s_1] \cdot D = C_1'[D, s_1]$ and $C_2'[s_2, B] \cdot s = C_2'[s_2, Bs]$ this can be written as

$$t = C \cdot C_1'[D, s_1] \cdot C_2'[s_2, Bs].$$

Hence,

$$\begin{aligned} t_1 \quad := \quad \mathrm{hs}(t, u_1x', u_2x') &= C \cdot C_1'[D, Bs] \cdot C_2'[s_2, s_1] \\ &= C \cdot C_1'[DC_2'[s_2, s_1], Bs]. \end{aligned}$$

The tree t_1 is a k-guarded horizontal swap of t.

Now we want to swap s and $DC_2'[s_2, s_1]$ in t_1. The node corresponding to the root of the tree s in the decomposition of t_1 is $u_1x'y$ where y is the hole of B, i.e. $v_2 = u_2x'y$. The node corresponding to the root of $DC_2'[s_2, s_1]$ is v_1. The tree $DC_2'[s_2, s_1]$ is k-similar to $DC_2'[s_2, Bs]$ which is the subtree below v_1 in t. Since v_1 and v_2 are k-similar in t, we obtain that $DC_2'[s_2, s_1]$ is also k-similar to s. Altogether, we obtain that $u_1x'y$ and v_1 are k-similar in t_1. We consider the following horizontal swap which, by the previous argument, is k-guarded:

$$\begin{aligned} t_2 \quad := \quad \mathrm{hs}(t_1, v_1, \tilde{u}_1x'y) &= C \cdot C_1'[s, BDC_2'[s_2, s_1]] \\ &= C \cdot C_1'[x \hookleftarrow s] \cdot BD \cdot C_2'[x \hookleftarrow s_2] \cdot s_1 \end{aligned}$$

We want to swap $C_1'[x \hookleftarrow s]$ and $C_2'[x \hookleftarrow s_2]$ in t_2. Note that the hole of both these contexts is x'. In t_2, the root of $C_1'[x \hookleftarrow s]$ corresponds to u_1 and the hole corresponds to u_1x'. Let z be the hole of D, i.e. $u_2 = v_1z$. Then, in t_2, the root of $C_2'[x \hookleftarrow s_2]$ corresponds to $u_1x'yz$ and the hole corresponds to $u_1x'yzx'$. We argue that u_1 and $u_1x'yz$ are $(k+\ell)$-similar in t_2 as follows. The subtree below $u_1x'yz$ in t_2 is $C_2'[s_2, s_1]$ and the subtree below u_1 is $C_1'[s, BDC_2'[s_2, s_1]]$. Recall that C_1' and C_2' are $(k+\ell)$-similar and that the holes x and x' of these contexts are at distance ℓ from the root. Recall that s and s_2 are k-similar. We have observed above that $DC_2'[s_2, s_1]$ and s are k-similar and we know that Bs and s_1 are k-similar. Hence, $BDC_2'[s_2, s_1]$ and s_1 are k-similar. Altogether, we obtain that $u_1x'yz$ and u_1 are $(k+\ell)$-similar in t_2.

Consider the vertical swap:

$$\begin{aligned}\mathrm{vs}(t_2, u_1, u_1x', u_1x'yz, u_1x'yzx') &= C \cdot C_2'[x \hookleftarrow s_2] \cdot BD \cdot C_1'[x \hookleftarrow s] \cdot s_1 \\ &= C \cdot C_2'[s_2, B] \cdot D \cdot C_1'[x' \hookleftarrow s_1] \cdot s \\ &= CC_2DC_1s = \mathrm{vs}(t, u_1, v_1, u_2, v_2).\end{aligned}$$

From the discussion above, it follows that this swap is k-trivial. Hence, we are done. □

We now proceed with the proof of Lemma 4.6.6.

Proof of Lemma 4.6.6. Suppose that L is closed under guarded horizontal swaps. By Lemma 4.6.4, we know that L is also closed under trivial vertical swaps. Let $k_0 \in \mathbb{N}$ witness the closure of L under both kinds of swaps. Let t be a tree, let $k \geqslant k_0$, and let $t' := \mathrm{vs}(t, u_1, v_1, u_2, v_2)$ be a k-short vertical swap. We show that either $t = t'$ or t' can be obtained from t by a sequence of swaps which are either k-trivial vertical swaps or k-guarded horizontal swaps, whence $t' \in L$.

Let $\ell_1 := \mathrm{dist}(u_1, v_1)$, $d := \mathrm{dist}(v_1, u_2)$, and $\ell_2 := \mathrm{dist}(u_2, v_2)$. Let CC_1DC_2s be the standard decomposition of t. Hence, $t' = CC_2DC_1s$.

We proceed by induction on $\ell := \ell_1 + d + \ell_2 = \mathrm{dist}(u_1, v_2)$ which we call the *length* of the swap. If $\ell_1 + \ell_2 = 0$, we have $t = t'$ and we are done. For the induction step, assume that $\ell_1 + \ell_2 > 0$. In the following, we discuss the case where $\ell_2 \geqslant \ell_1$; the case where $\ell_2 < \ell_1$ follows by symmetric reasoning.

Case I: $\ell_1 > 0$.

In this case, $u_1 \neq v_1$ and hence there is an non-empty $x \in \Pi^{\ell_1}$ such that $u_1x = v_1$.

If $u_2x \parallel v_2$, then $\mathrm{vs}(t, u_1, v_1, u_2, v_2)$ satisfies the conditions of Lemma 4.6.7. From that lemma, we know that t' can be obtained from t using only k-trivial vertical swaps and k-guarded horizontal swaps and we are done.

In the following, we consider the case that $u_2x \trianglelefteq v_2$. Since u_1, u_2 are $(k+\ell)$-similar and $|x| = \ell_1 \leqslant \ell$, we know that the swap $t'' := \mathrm{vs}(t, u_1, v_1, u_2, u_2x)$ is k-trivial. Let $C_2 = C_{2,1}C_{2,2}$ for $C_{2,1} := t[u_2, u_2x)$ and $C_{2,2} := t[u_2x, v_2)$ and note that

$$t'' = CC_{2,1}DC_1C_{2,2}s.$$

In t'', the node $v_1 = u_1x$ corresponds to the root of D and u_2x and v_2 correspond to the roots of $C_{2,2}$ and s, respectively.

Consider the following swap:

$$\mathrm{vs}(t, v_1, v_1, u_2x, v_2) = CC_{2,1}C_{2,2}DC_1s = t'.$$

We want to apply induction to show that we can obtain t' from t'', and hence from t, using k-guarded horizontal swaps and k-trivial vertical swaps. The length of the swap is $\mathrm{dist}(v_1, v_2) = \ell - \ell_1$. Since $\ell_1 > 0$, this is strictly smaller than ℓ. The nodes $v_1 = u_1x$ and u_2x are $(k + \ell - \ell_1)$-similar and v_1 and v_2 are $(k+\ell)$-similar. Hence, the swap is k-short and we can apply induction as desired.

Case II: $\ell_1 = 0$.

In this case, $u_1 = v_1$ and hence C_1 is the empty context. That is, we have $t = CDC_2s$ and we want to exchange D and C_2 to obtain

$$t' = CC_2Ds.$$

We distinguish further cases depending on d and ℓ_2.

Case II.a: $d = 0$ or $\ell_2 = 0$.

In both cases, we are finished without further ado: in the first case, $t = CC_2s = t'$ and in the second case $t = CDs = t'$.

Case II.b: $0 < d \leqslant \ell_2$.

By our initial assumption, we know that u_1 and u_2 are $(k+\ell)$-similar. Since $u_1 = v_1$ and since v_1 and v_2 are $(k+\ell)$-similar in t, we know that u_2 and v_2 are also $(k+\ell)$-similar. Hence, the swap $\mathrm{vs}(t, v_1, u_2, u_2, v_2) = t'$ is k-short. Its length is $\mathrm{dist}(v_1, v_2) = \ell$. Since $\mathrm{dist}(v_1, u_2) = d > 0$ and $\ell_2 \geqslant d$, we can proceed as in Case I.

Case II.c: $d > \ell_2 > 0$.

Let $x \in \Pi^{\ell_2}$ be such that $v_2 = u_2x$.

Case II.d.a: $v_1x \parallel u_2$.

In this case, $\mathrm{vs}(t, v_1, u_2, u_2, v_2) = t'$ satisfies the condition of Lemma 4.6.7. That is, we have $\mathrm{dist}(u_2, v_2) = \ell_2$ and $v_1 = u_1$ is $(k+\ell_2)$-similar to u_2 by the initial assumption of this lemma, and u_2 is k-similar to v_2 since — also by the initial assumption — v_1 is at least k-similar to v_2. According to Lemma 4.6.7, t' can be obtained from t using only k-guarded horizontal swaps and k-trivial vertical swaps. Thus, we are done with this case.

Case II.d.b: $v_1x \trianglelefteq u_2$.

We split D into two contexts $D_1 := t[v_1, v_1x)$ and $D_2 := t[v_1x, u_2)$ such that $D = D_1D_2$. Consider the swap

$$t'' \ := \ \mathrm{vs}(t, v_1, v_1x, u_2, u_2x) \ = \ CC_2D_2D_1s.$$

Since $v_1 = u_1$, this swap is clearly k-trivial.

Finally, to end the proof, we want to exchange D_2 and D_1 in t'' to obtain t'. Let y denote the hole of D_2. The following vertical swap achieves our goal.

$$\mathrm{vs}(t'', v_1x, u_2, u_2, v_2) \ = \ CC_2D_1D_2s \ = \ t'.$$

Note that the length of this swap is $\mathrm{dist}(v_1x, v_2) = d$. We consider the similarity of the relevant nodes to show that the swap is k-short. First, we show that v_1x and u_2 are $(k+d)$-similar. The subtree below v_1x in t'' is D_2D_1s. This tree is $(k+\ell)$-similar to the tree D_2C_2s which is the subtree below v_1x in t. The subtree below u_2 in t'' is D_1s. This tree is $(k+\ell)$-similar to the tree C_2s which is the subtree below u_2 in t. The subtree below v_2 in t'' is s which is also the subtree below v_2 in t. Hence, it remains to show that v_1x, u_2, and v_2 are mutually $(k+d)$-similar in t. We know that u_2 is $(k+\ell)$-similar to $u_1 = v_1$ in

t. On the one hand, together with the $(k+\ell)$-similarity of v_1 and v_2, this implies that u_2 is $(k+d)$-similar to v_2. Furthermore, $u_2x = v_2$ and v_1x are $(k+\ell-\ell_2)$-similar in t, where $\ell - \ell_2 = d$. Hence, by induction, t' can be obtained from t'', and hence from t, using only k-trivial and k-guarded horizontal swaps.

□

Definition 4.6.8 (gaping)**.** A vertical swap $\mathrm{vs}(t, u_1, v_1, u_2, v_2)$ is *k-gaping* if it is k-guarded and $\mathrm{dist}(v_1, u_2) \geqslant k$.

Lemma 4.6.9. *If a tree language L is closed under strongly guarded vertical swaps and closed under trivial vertical swaps, then it is also closed under gaping vertical swaps.*

Proof. Let $k_0 \in \mathbb{N}$ be a parameter witnessing the closure under strongly guarded vertical swaps and under trivial vertical swaps.

Let $k \geqslant k_0$, let t be a tree, and let $t' := \mathrm{vs}(t, u_1, v_1, u_2, v_2)$ be a $(4k+2)$-gaping swap of t. We show that t' can be obtained from t using both strongly k-guarded vertical swaps and $(2k+1)$-trivial vertical swaps, which implies that $t' \in L$.

Let CC_1DC_2s be the standard decomposition of t. Let $x \in \Pi^{2k+1}$ be such that $v_1x \trianglelefteq u_2$. Note that, by $(2k+1)$-similarity of v_1 and v_2, the node v_2x belongs to t. We have $D = D_1D_2$ for $D_1 := t[v_1, v_1x)$ and $D_2 := t[v_1x, u_2)$. We also have $s = Es'$ for $E := t[v_2, v_2x)$ and $s' := t_{|v_2x}$. Thus

$$t'' \;:=\; \mathrm{vs}(t, v_1, v_1x, v_2, v_2x) \;=\; CC_1 \cdot E \cdot D_2C_2 \cdot D_1 \cdot s'.$$

Since $\mathrm{dist}(v_1, v_1x) = \mathrm{dist}(v_2, v_2x) = 2k+1$ and v_1 and v_2 are $2(2k+1)$-similar, the swap t'' is $(2k+1)$-trivial. In the tree t'', the nodes u_1 and u_2 correspond to the roots of C_1E and C_2D_1 and v_1x and v_2x correspond to the roots of D_2 and s'. The nodes u_1 and u_2 are clearly k-similar. For the next step, we need to verify that the k-spheres of v_1x and v_2x in t'' are isomorphic. Note that the k-spheres of v_1x and v_2x in t are isomorphic to the k-spheres of the node x in D_1D_2 and Es', respectively. Since D_1 and E are $(3k+1)$-similar, $|x| = 2k+1$, and D_2 and s' are k-similar, these k-spheres are isomorphic. This implies that the k-spheres of x in ED_2 and in D_1s' are isomorphic. Hence, the k-spheres of v_1x and v_2x in t'' are isomorphic. Consider the vertical swap

$$\begin{aligned} \mathrm{vs}(t'', u_1, v_1x, u_2, v_2x) \;:&=\; C \cdot C_2D_1 \cdot D_2 \cdot C_1E \cdot s' \\ &=\; CC_2DC_1s \;=\; t'. \end{aligned}$$

Obviously, $\mathrm{dist}(u_1, v_1x), \mathrm{dist}(u_2, v_2x) \geqslant 2k+1$. Since t' is, in particular, $(3k+2)$-gaping, we obtain $v_1x \trianglelefteq u_2$ and $\mathrm{dist}(v_1x, u_2) \geqslant \mathrm{dist}(v_1, u_2) - (2k+1) \geqslant 3k+2-2k-1 = k+1$. Altogether, the previous swap is strongly k-guarded. □

Definition 4.6.10 (long)**.** A vertical swap $\mathrm{vs}(t, u_1, v_1, u_2, v_2)$ is *k-long* if it is $2k$-guarded and if $\mathrm{dist}(u_1, v_1) \geqslant k$ and $\mathrm{dist}(u_2, v_2) \geqslant k$.

Note that a swap which is not k-long is not necessarily k-short.

Lemma 4.6.11. *If a tree language is closed under trivial and under gaping vertical swaps, then it is also closed under long vertical swaps.*

Proof. Let k_0 be the parameter witnessing the closure of L under trivial and gaping vertical swaps. Let t be a tree, let $k \geqslant k_0$, and let $t' := \mathrm{vs}(t, u_1, v_1, u_2, v_2)$ be a k-long vertical swap of t. We show that t' can be obtained from t by k-trivial or k-gaping vertical swaps which implies that $t' \in L$.

From now on, we assume that the given vertical swap of t is not k-gaping; otherwise we are done. Hence, we have $\mathrm{dist}(v_1, u_2) < k$. Let CC_1DC_2s be the standard decomposition of t for the given nodes. Let $x \in \Pi^*$ with $|x| < k$ such that $u_2 = v_1x$. Let $\tilde{u}_2 := v_2x$. By $2k$-similarity of v_1 and v_2 in t, the nodes u_2 and $\tilde{u}_2$ are k-similar. Let $D' := t[v_2, \tilde{u}_2)$ and let $s' := t_{|\tilde{u}_2}$. Consider the vertical swap

$$t'' := \mathrm{vs}(t, u_1, u_1, u_2, \tilde{u}_2) \;=\; C \cdot C_2D' \cdot C_1D \cdot s'.$$

and note that this swap is k-gaping since $\mathrm{dist}(u_1, u_2) \geqslant \mathrm{dist}(u_1, v_1) \geqslant k$ and u_1, u_2, and $\tilde{u}_2$ are all k-similar.

We want to swap D' and D. Let y be the hole of C_2. Then u_1y corresponds to the root of D' in t'' and u_1yx corresponds to the root of C_1. Let z be the hole of C_1. Thus u_1yxz corresponds to the root of D and u_1yxzx is the root of s' in t''. Note that D and D' are $2k$-similar, since v_1 and v_2 are $2k$-similar in t and both have the same hole x. Since u_1 and $\tilde{u}_2$ are k-similar in t, we know that C_1 and s' are also k-similar. Hence, the subtrees $D'C_1Ds'$ and Ds' below u_1y and u_1yxz in t'' are $(k+|x|)$-similar. Hence, the following swap is k-trivial and we are done:

$$\mathrm{vs}(t'', u_1y, u_1yx, u_1yxz, u_1yxzx) \;=\; CC_1 \cdot D \cdot C_2 \cdot D' \cdot s' = t'.$$

□

So far we have considered vertical swaps which are *balanced* in the sense that either the lengths of (the root-hole paths) of the contexts C_1 and C_2 are both shorter than the height of the subtrees which are isomorphic according to the guardedness assumptions (k-short swaps), or both are longer (k-long swaps), or D is sufficiently long and hence we have enough room to compensate for the difference in the lengths (k-gaping swaps). The next kind of vertical swaps which we consider are *unbalanced* in the sense that either the length of C_1 is much longer than the length of DC_2 or the length of C_2 is much longer than the length of C_1D.

Definition 4.6.12 (unbalanced)**.** A vertical swap $\mathrm{vs}(t, u_1, v_1, u_2, v_2)$ is k-*unbalanced* if it is $4k^2$-guarded, and either $\mathrm{dist}(u_1, v_1) > 4k^2$ and $\mathrm{dist}(v_1, v_2) \leqslant 2k$, or $\mathrm{dist}(u_2, v_2) > 4k^2$ and $\mathrm{dist}(u_1, u_2) \leqslant 2k$.

Lemma 4.6.13. *If L is a tree language which is closed under long vertical swaps, then L is also closed under unbalanced vertical swaps.*

Proof. Let k_0 be the parameter witnessing the closure under long vertical swaps. Let t be a tree and let $t' := \mathrm{vs}(t, u_1, v_2, u_2, v_2)$ be a k-unbalanced vertical swap. We show that either $t = t'$ or t' can be obtained from t by k-long vertical swaps. For each $k \geqslant k_0$, this implies that $t' \in L$.

Let $K := 4k^2$. Let CC_1DC_2s be the standard decomposition of t. Let $\ell_1 := \mathrm{dist}(u_1, v_1)$, $d := \mathrm{dist}(v_1, u_2)$, and $\ell_2 := \mathrm{dist}(u_2, v_2)$. There are two cases to consider.

Case I: $\ell_1 \geqslant K$ and $\ell_2 + d \leqslant 2k$.

Let x be the hole of C_2, i.e. $v_2 = u_2x$, and note that $|x| = \ell_2$. Let y be the hole of D, i.e. $u_2 = v_1y$, and note that $|y| = d$. If $t = t'$, we are done. If $\ell_2 + d = |yx| = 0$, then DC_2 is the empty context and hence $t = CC_1s = t'$. That is, we can assume that $t \neq t'$ and hence $\ell_2 + d \geqslant 1$. This implies $k \geqslant 1$.

Since v_1 is K-similar to v_2, we obtain that v_2y is a node of t which is $(K - d)$-similar to u_2 and hence to u_1. Then v_2yx is $(K - (d + \ell_2))$-similar to v_2. We repeat this argument for $m := \lfloor \frac{K-2k}{d+\ell_2} \rfloor \geqslant \frac{2k(2k-1)}{2k} \geqslant k$ times and we obtain that the node $\tilde{v}_2 := v_2(yx)^m$ is $2k$-similar to v_1 and v_2. Furthermore, $\mathrm{dist}(v_2, \tilde{v}_2) \geqslant m \cdot |yx| \geqslant k$ since $|yx| \geqslant 1$. Hence, the swap $\mathrm{vs}(t, u_1, v_1, u_2, \tilde{v}_2)$ is k-long. Let $s = D's'$ where $D' := t[v_2, \tilde{v}_2)$ and $s' := t_{|\tilde{v}_2}$. Then

$$t_1 \ := \ \mathrm{vs}(t, u_1, v_1, u_2, \tilde{v}_2) \ := \ C \cdot C_2D' \cdot D \cdot C_1 \cdot s'.$$

Note that the node $u_1' := u_1x$ in t_1 corresponds to the root of D' and that it is $2k$-similar to v_2 in t. Note that the node $v_1' := u_1'(yx)^m$ corresponds to the root of D in t_1 and that it is $2k$-similar to $\tilde{v}_2$ in t and hence to v_2 in t. Furthermore, the root of s' corresponds to $v_2' := v_1'yz$ in t_1, where z is the hole of C_1. Further, v_2' is $2k$-similar to $\tilde{v}_2$ in t and hence to v_2 in t. We can apply the vertical swap

$$\mathrm{vs}(t_1, u_1', v_1', v_1', v_2') \ := \ CC_2DC_1D's' \ = \ t'.$$

This is is a k-long swap since $\mathrm{dist}(u_1', v_1') \geqslant m \geqslant k$ and $\mathrm{dist}(v_1', v_2') \geqslant \ell_1 \geqslant k$.

Case II: $\ell_2 \geqslant K$ and $d + \ell_1 \leqslant 2k$.

Let x be the hole of C_1, i.e. $v_1 = u_1x$. Note that $|x| = \ell_1$. Let y be the hole of D, i.e. $u_2 = v_1y$, and note that $|y| = d$. Analogously to the first case of the proof, we can assume that $t \neq t'$ and $d + \ell_1 = |yx| \geqslant 1$. Now v_2y is a node of t which is $(K - d)$-similar to u_2 and hence to u_1. Thus, v_2yx is a node of t which is $(K - (d + \ell_1))$-similar to v_1 and hence to v_2. We repeat this argument for $m := \lfloor \frac{K-2k}{d+\ell_1} \rfloor \geqslant \frac{2k(2k-1)}{2k} \geqslant k$ times and we obtain that the node $\tilde{v}_2 := v_2(yx)^m$ is $2k$-similar to v_1. Furthermore, $\mathrm{dist}(v_2, \tilde{v}_2) \geqslant m \cdot |yx| \geqslant k$ since $|yx| \geqslant 1$. Hence, the swap $\mathrm{vs}(t, v_1, v_2, v_2, \tilde{v}_2)$ is k-long. Let $s = D's'$ where $D' := t[v_2, \tilde{v}_2)$ and $s' := t_{|\tilde{v}_2}$. Then

$$t_1 \ := \ \mathrm{vs}(t, v_1, v_2, v_2, \tilde{v}_2) \ := \ CC_1D'DC_2s'.$$

Note that here the node $v_1' := v_1(yx)^m$ corresponds to the root of D, that $u_2' := v_1'y$ corresponds to the root of C_2, and that $v_2' := u_2'z$ corresponds to the root of s', where z is the hole of C_2. Furthermore, u_1 and u_2' are $2k$-similar and v_1' and v_2' are $2k$-similar in t_1. Then

$$\mathrm{vs}(t, u_1, v_1', u_2', v_2') \ := \ CC_2DC_1D's' = t'$$

is a k-long swap.

□

The proof of Theorem 4.6.2 can now be completed as follows.

Proof of Theorem 4.6.2. The implication from closure under guarded swaps to closure under strongly guarded vertical swaps and guarded horizontal swaps is trivial.

For the reverse implication, we assume that L is a tree language which is closed under guarded horizontal swaps and strongly guarded vertical swaps. Combining Lemma 4.6.6, Lemma 4.6.9, Lemma 4.6.11, and Lemma 4.6.13, we obtain that L is closed under short, under gaping, under long, and under unbalanced vertical swaps. Let k be a parameter witnessing the closure of L under all four kinds of swaps.

We show that L is closed under K-guarded vertical swaps for $K := 8(k+1)^2$. Consider a tree t and an K-guarded vertical swap $t' = \mathrm{vs}(t, u_1, v_1, u_2, v_2)$ of t. If t' is $4k^2$-short, or $(k+1)$-gaping, or $(k+1)$-long, we obtain that $t' \in L$ and we are done. Assume that none of this applies to t' and recall the definitions of the different kinds of vertical swaps. Since the vertical swap is not $(k+1)$-gaping but at least $(k+1)$-guarded, we have

$$\mathrm{dist}(v_1, u_2) \leqslant k. \tag{4.5}$$

Hence, and since the vertical swap is not $4k^2$-short but at least $(2(4k^2)+k)$-guarded, we must have

$$\mathrm{dist}(u_1, v_1) > 4k^2 \quad \text{or} \quad \mathrm{dist}(u_2, v_2) > 4k^2.$$

Since the vertical swap is not $(k+1)$-long but at least $2(k+1)$-guarded, it cannot happen that both distances are large at the same time, i.e.

$$\mathrm{dist}(u_1, v_1) \leqslant k \quad \text{or} \quad \mathrm{dist}(u_2, v_2) \leqslant k.$$

This leads to two cases. If $\mathrm{dist}(u_1, v_1) > 4k^2$, we must have $\mathrm{dist}(u_2, v_2) \leqslant k$ and thus $\mathrm{dist}(v_1, v_2) \leqslant 2k$ by (4.5). If $\mathrm{dist}(u_2, v_2) > 4k^2$, we must have $\mathrm{dist}(u_1, v_1) \leqslant k$ and thus $\mathrm{dist}(u_1, u_2) \leqslant 2k$ by (4.5). In either case, the vertical swap is k-unbalanced and hence $t' \in L$. This finishes the proof. □

4.6.2 Applications

Theorem 4.6.2 removes much of the combinatorial burden associated with applications of guarded swaps. In particular, labour intensive case distinctions due to the possible overlappings of the k-spills of the nodes involved in a k-guarded vertical swap are obviated by it. For some results involving swaps, the published proofs considered only the case with no overlappings of k-spills. Our Theorem 4.6.2 can, in particular, be seen as providing the necessary details to complete these proof sketches.

As an example, consider the following result.

Theorem 4.6.14 ([BS09a]). *A regular language L of words is closed under idempotent-guarded swaps iff it is closed under guarded swaps.*

A word language is *closed under idempotent-guarded swaps* if for all words e, f which are idempotent with respect to L and all words $u, \underline{s}, v, \underline{s}', w$, the words $ue\underline{s}fve\underline{s}'fw$ and $ue\underline{s}'fve\underline{s}fw$ are L-equivalent. This closure property, phrased as a property of syntactic monoids, is part of the characterisation of FO on words without the prefix order [BP89]. In the part of the proof of Theorem 4.6.14 where the implication from closure under idempotent-guarded swaps to closure under guarded swaps is established, it is shown that each k-guarded vertical swap $\mathrm{vs}(w, x, y, x', y')$ of a word w, for sufficiently large k depending on the language L, can be obtained from w by an idempotent-guarded swap. Only two special cases are considered as examples, stating that the other cases are "handled similarly". In particular, the case "in which y is not in the k-spill of x, x' is not in the k-spill of y, and y' is not in the k-spill of y'" is fully treated. Since this is always true in a strongly k-guarded swap, the proof actually establishes that each strongly k-guarded swap can be obtained using only idempotent-guarded swaps. Using our Theorem 4.6.2, this suffices to conclude that L is closed under guarded swaps.

A similar situation occurs in the proof of [BS09a, Lemma 13]. Furthermore, several results are only proved for guarded horizontal swaps and it is only stated that the case of vertical swaps is "similar"; cf. e.g. [BS09a, Lemma 11]. This seems only sufficiently justified if one considers strongly guarded swaps instead of guarded swaps. This situation has also lead to a significant gap in the proof of [BS09b, Theorem 4.2] where it is shown that all order-invariantly FO-definable tree languages are closed under guarded swaps. A proof for the case of horizontal swaps is given, but the case of vertical swaps is only summarised roughly and it is claimed that the proof can be carried out in analogy to the proof for horizontal swaps. But in the case of vertical swaps, the application of [BS09b, Lemma 4.6] is unjustified, since the lemma presumes that the elements for which it is applied have isomorphic k-spheres although only k-similarity of the nodes from which these elements derive has been assumed. It is not at all obvious how the requirement of the lemma could be satisfied from the given assumption. By our Theorem 4.6.2, it suffices to show that all order-invariant tree languages are closed under strongly guarded vertical swaps. With this stronger assumption, the requirements for the application of the lemma are satisfied and the proof can indeed be carried out analogously to the case for horizontal swaps.

4.7 Addition-invariant FO

In the following section, we are concerned with our second main result which establishes a connection between the $\mathrm{FO}_{\mathrm{card}}$-definable languages which we investigated in Section 4.4 and addition-invariance.

4.7.1 Addition-invariantly definable tree languages

In the following, we assume that the label alphabet Σ and the maximum rank r of all trees are fixed, unless we state something else. We say that a sentence is *addition-invariant* if it is addition-invariant on the class of all Σ-labelled trees of rank r. For this section, we let

$+$-inv-FO $:= +$-inv-FO$[\sigma_{\Sigma,r}]$ and we say that a tree language is $+$-inv-FO-definable if the corresponding class of $\sigma_{\Sigma,r}$-structures is $+$-inv-FO-definable on the class of all Σ-labelled trees of rank r.

The following theorem is our second main result which generalises a result of [SS10] from words to trees:

Theorem 4.7.1. *Let L be a regular tree language. The following statements are equivalent:*

1. *L is* $+$-inv-FO*-definable,*
2. *L is closed under transfer and guarded swaps,*
3. *L is* $\mathrm{FO}_{\mathrm{card}}$*-definable.*

The equivalence of statements (3) and (2) was proved in Theorem 4.4.7. The implication from statement (3) to statement (1) is easy to prove. Consider a regular tree language which is definable by an $\mathrm{FO}_{\mathrm{card}}$-sentence φ. For each $m \in \mathbb{N}^+$ and each $r \in [m]$, the set of trees of size r modulo m is defined by an $+$-inv-FO-sentence $\varphi_{r,m}$ of the following shape:

$$\exists x\, \exists y\, \exists z\ (y = m \cdot x\ \wedge\ z = y{+}r\ \wedge\ \max(z)).$$

The terms $z = y{+}r$ and $m \cdot x$, where r, m are constant, are shorthand for corresponding formulae over the relational signature $\{+, <\}$ and $\max(z)$ is shorthand for $\forall z'\ z'{<}z \vee z = z'$. The sentence obtained from φ by replacing each cardinality predicate $\mathrm{C}_{r,m}$ by the sentence $\varphi_{r,m}$ is an addition-invariant sentence which defines L.

For the remainder of this section, we will be occupied by the proof of the implication from (1) to (2). That is, we show that $+$-inv-FO-definability implies closure under guarded swaps and transfer.

4.7.2 Closure under guarded horizontal swaps

In this section, we prove the following lemma.

Lemma 4.7.2. *Let L be a regular tree language. If L is definable by an* $+$-inv-FO*-sentence, then L is closed under guarded horizontal swaps.*

We start with a rough overview of the proof. We use this to introduce necessary definitions and lemmas. Consider a tree t and any of its horizontal swaps $\mathrm{hs}(t, u, v)$ which is k-guarded for a sufficiently large k. For technical reasons that become apparent in the proof below, we partition the domains of the considered trees into parts of the same size e and we define a structure whose elements are the parts of this partition. The relations of the structure contain everything that is necessary to recover the original trees by a first-order interpretation.

For the formal definition of these structures which we call *e-block decompositions*, we need some notation. For $S \subseteq \Pi^*$ and $i \in [|S|]$, we let $S@i$ (read "S at i") denote the element of S with index i in the lexicographic order on S. For a tree t, we let $t@i := \mathrm{dom}(t)@i$ and for a context C we let $C@i := s@i$ where s is the underlying tree of C.

Definition 4.7.3 (e-block decomposition)**.** Let $e \in \mathbb{N}^+$ and let t be a tree. Let $n := |t|$, let $m := \lfloor \frac{n}{e} \rfloor$ and let $\rho := n \operatorname{rem} e$. For each $j \in [1,m]$, the *j-th block* of t is the set

$$\mathrm{B}_j^t := \{t@i : i \in [(j-1)e, je-1]\}.$$

If $\rho \geqslant 1$, the *remainder block* of t is the set

$$\mathrm{R}^t := \{t@i : i \in [me+1, n]\}$$

which contains the last ρ elements of t in the lexicographical order.

The *e-block decomposition* of t is the relational structure $\mathfrak{D}_t$ whose universe is

$$D_t := \{\mathrm{B}_j^t : j \in [1,m]\}$$

and which for all $i, j \in [e]$ and $k \in \Pi$ contains relations

$$S_{i,j,k}^{\mathfrak{D}_t} := \{(\mathrm{B}, \mathrm{B}') \in D_t^2 : \mathrm{B}'@j = u, \mathrm{B}@i = v, u = vk\}$$

that encode the successor relations in t between the nodes contained in the blocks, and, for all $i \in [e]$ and $a \in \Sigma$, further relations

$$C_{i,a}^{\mathfrak{D}_t} := \{\mathrm{B} \in D_t : t(\mathrm{B}@i) = a\}$$

that encode the labels of the elements of the blocks in t.

Furthermore, the structure contains unary relations

$$K_{i,j,k}^{\mathfrak{D}_t} := \{\mathrm{B} \in D_t : u = \mathrm{R}^t@j, v = \mathrm{B}@i, u = vk\}$$

which encode the information that the j-th element of the remainder block is the k-successor of the i-th element of a block. Note that there are no successor relations in the other direction, i.e. all successors of nodes of R^t in t also belong to R^t, since each successor of a node is lexicographically larger than the node.

We construct the disjoint union $\mathfrak{D}$ of the e-block decompositions of the two subtrees $t_{|u}$ and $t_{|v}$ of t which are swapped in $\operatorname{hs}(t,u,v)$, for some fitting value of e. Then we apply the following lemma to obtain two linear orders of $\mathfrak{D}$ where the relative position of the parts corresponding to the roots of $t_{|u}$ and $t_{|v}$ is different, (i.e. u comes before v in the first order and v comes before u in the second order) and such that the two ordered expansions of $\mathfrak{D}$ satisfy the same first-order sentences of at most some given quantifier-rank.

Lemma 4.7.4 (implicit in [GS00])**.** *Let $x, q' \in \mathbb{N}$, and let σ be a signature. There exists $k' \in \mathbb{N}$ such that for each finite σ-structure $\mathfrak{D}$ and all x-tuples $\bar{a}$ and $\bar{b}$ of $\mathfrak{D}$ with isomorphic k'-spheres, there exist linear orders $<_1$ and $<_2$ of D, whose initial elements are respectively $\bar{a}\bar{b}$ and $\bar{b}\bar{a}$, such that $(\mathfrak{D}, <_1) \equiv_{q'} (\mathfrak{D}, <_2)$.*

This lemma is the technical core of the proof of the Gaifman-locality of $<$-invariant-FO (cf. [GS00]). It was also used in [BS09b] to show that order-invariantly definable tree languages are closed under swaps.

We use a technique which allows to transfer the $\equiv_{q'}$-equivalence of the two ordered structures to $\equiv_{q''}$-equivalence of two expansions of the structure of non-negative integers with addition, each one containing an embedded copy of one of the ordered structures.

To state the lemma, we need the following notation. Let σ be a signature. For each σ-structure $\mathfrak{A}$, we write $\sigma^{\mathfrak{A}}$ for the set of relations of $\mathfrak{A}$. Consider σ-structures $\mathfrak{A}, \mathfrak{B}$ and a mapping α from the universe of $\mathfrak{A}$ to the universe of $\mathfrak{B}$. For a relation $R \in \sigma^{\mathfrak{A}}$ of arity m, we define $\alpha(R) := \{(\alpha(a_1), \ldots, \alpha(a_m)) : (a_1, \ldots, a_m) \in R\}$. For $\sigma^{\mathfrak{A}} = \{R_1, \ldots, R_n\}$, let $\alpha(\sigma^{\mathfrak{A}}) := \{\alpha(R_1), \ldots, \alpha(R_n)\}$.

Lemma 4.7.5 ([Sch07]). *Let $q'', h, e \in \mathbb{N}^+$ and let σ be a signature. There exists an infinite set $P := \{p_1 < p_2 < p_3 \ldots\} \subseteq \mathbb{N}$ with $p_1 > h$ and $p_i \equiv h\,(mod\,e)$, for all $i \in \mathbb{N}^+$, and a number q' such that the following is true for all finite σ-structures $\mathfrak{M}$ and all linear orders $<_1$ and $<_2$ on M: if $(\mathfrak{M}, <_1) \equiv_{q'} (\mathfrak{M}, <_2)$, then $(\mathbb{Z}, +, P, \alpha_1(\sigma^M)) \equiv_{q''} (\mathbb{Z}, +, P, \alpha_2(\sigma^M))$, where α_i is a map taking the j-th node of M according to $<_i$ to p_j, for all $i \in [1, 2]$ and $j \in \mathbb{N}^+$.*

Lemma 4.7.5 is an immediate consequence of [Sch07, Proposition 6.11] and has also been used in the results of [SS10].

The last step which constitutes the main part of the proof of Lemma 4.7.2 consists in a first-order interpretation which uses the information about the block decompositions and the addition relation in the structures of Lemma 4.7.5 to simulate addition expansions of trees t and $\mathrm{hs}(t, u, v)$ (or rather trees which are equivalent with respect to the given tree language). The interpretation will depend on the parts of t which are not contained in the block decompositions (e.g. the subtree above u and v, the remainder block) but from regularity of L we obtain that, up to L-equivalence, there are only finitely many choices to consider for these parameters. The $\equiv_{q''}$-equivalence of the structures of Lemma 4.7.5 — where q'' will be the maximum quantifier-rank of the formula obtained by rewriting the defining formula φ for L according to the interpretation, for any given choice of the parameters — will then imply that the addition expansions of the trees must agree on φ. By addition-invariance, the trees are L-equivalent and we are done.

Proof of Lemma 4.7.2. Let φ be an $+$-inv-FO sentence defining the language L. Let Q be the state set of a minimal deterministic tree automaton recognising the language L. We want to show that L is closed under k-guarded horizontal swaps, for a number k that will be fixed later on. To this end, consider a tree $t \in L$ with incomparable k-similar nodes u and v. We must show that $\mathrm{hs}(t, u, v) \in L$. Let $t_1 := t_{|u}$ and $t_2 := t_{|v}$, so that $t = C[t_1, t_2]$ and $\mathrm{hs}(t, u, v) = C[t_2, t_1]$, for some 2-context C.

By k-similarity, we may assume that the trees t_1 and t_2 have height at least k: if $\mathrm{height}(t_1), \mathrm{height}(t_2) \leqslant k$, then both trees are equal and we are done; assuming $\mathrm{height}(t_1) < k$ and $\mathrm{height}(t_2) \geqslant k$ contradicts k-similarity.

Let $k' \in \mathbb{N}$ be a number whose value will be fixed later on. If we let $k > \kappa k' + |Q^Q|$, we know that there exists an idempotent context in a subtree of t_1 whose root has depth larger than $\kappa k'$. This means, $t_1 = DEt_1'$ for a tree t_1' and two contexts D, E such that E is idempotent. Moreover, D has the same $\kappa k'$-prefix as t_1, and is thus $\kappa k'$-similar to t_2. We may assume that $e := |E| \leqslant \kappa$. Otherwise, by definition of κ, we could exchange E for a context of size at most e without affecting membership of t and $\mathrm{hs}(t, u, v)$ in L. Analogously, we can assume that the size of C is bounded by a constant which depends only on L, since, as for 1-contexts, there is a constant κ' such that for each 2-context C there is an L-equivalent 2-context C' with $|C'| \leqslant \kappa'$. Here, 2-contexts C, C' are L-equivalent if the trees $C[t_1, t_2]$ and $C'[t_1, t_2]$ are L-equivalent, for all trees t_1, t_2. We also assume that $|t_1'| \geqslant e$. Otherwise, we could prepend a copy of E to t_1' to achieve this without affecting membership of t and $\mathrm{hs}(t, u, v)$ in L.

Let $\mathfrak{D}_1 := \mathfrak{D}_{t_1}$ and $\mathfrak{D}_2 := \mathfrak{D}_{t_2}$ be e-block-decompositions of t_1 and t_2 as defined in Definition 4.7.3. Note that the signature β of e-block decompositions depends only on e and the fixed alphabet and rank of the trees. We take the disjoint union $\mathfrak{D}_1 \sqcup \mathfrak{D}_2$ of $\mathfrak{D}_1$ and $\mathfrak{D}_2$, i.e. the β-structure with the universe $(\mathfrak{D}_1 \times \{1\}) \cup (\mathfrak{D}_2 \times \{2\})$ where $R^{\mathfrak{D}_1 \sqcup \mathfrak{D}_2} := \{((a, i), (b, i) : (a, b) \in R^{\mathfrak{D}_i}, i \in \{1, 2\}\}$, for each $R \in \beta$. Let $\sigma := \beta \cup \{T_1, T_2, J_E\}$ where T_1, T_2, J_E are unary relation symbols which do not occur in β. We extend $\mathfrak{D}_1 \sqcup \mathfrak{D}_2$ to a σ-structure $\mathfrak{D}$ as follows:

- $J_E^{\mathfrak{D}}$ contains only the element of $\mathfrak{D}_1$ which corresponds to the block of t_1 which contains the parent of the hole of E.
- $T_i^{\mathfrak{D}}$ contains all elements which correspond to a block of $\mathfrak{D}_i$ which contains nodes which are parents of a node from the remainder block of $\mathfrak{D}_i$, for each $i \in \{1, 2\}$.

We want to apply Lemma 4.7.4 to the σ-structure $\mathfrak{D}$ with $x := 1$, $\bar{a} := (\mathrm{B}_0^{t_1}, 1)$, and $\bar{b} := (\mathrm{B}_0^{t_2}, 2)$ for a number q' whose value will be fixed below. Let k' be chosen according to Lemma 4.7.4. In order to apply Lemma 4.7.4, we need to ensure that $(\mathrm{B}_0^{t_1}, 1)$ and $(\mathrm{B}_0^{t_2}, 2)$ have isomorphic k'-spheres in $\mathfrak{D}$. Because the k-spills of the root of t_1 and the root of t_2 are isomorphic, it is plain to see that the k/e-spheres of $\mathrm{B}_0^{t_1}$ in $\mathfrak{D}_1$ and that of $\mathrm{B}_0^{t_2}$ in $\mathfrak{D}_2$ are isomorphic. Note that we have $k > \kappa k' \geqslant ek'$ and that the new unary relations in $\mathfrak{D}$ do not contain any elements from the k'-spheres of $(\mathrm{B}_0^{t_1}, 1)$ and $(\mathrm{B}_0^{t_2}, 2)$ in $\mathfrak{D}$. Hence, these spheres are isomorphic. By Lemma 4.7.4 we obtain two linear orderings $<_1$ and $<_2$ on $\mathfrak{D}$ such that $(\mathfrak{D}, <_1) \equiv_{q'} (\mathfrak{D}, <_2)$ and such that $(\mathrm{B}_0^{t_1}, 1)$ is the first element and $(\mathrm{B}_0^{t^2}, 2)$ is the second element according to $<_1$, whereas $(\mathrm{B}_0^{t_2}, 2)$ is the first element and $(\mathrm{B}_0^{t_1}, 1)$ is the second element according to $<_2$.

For $i \in \{1, 2\}$, let $n_i := \lfloor \frac{|t_i|}{e} \rfloor$ and let $\rho_i := |t_i| \,\mathrm{rem}\, e$. Let $n := |D|$. Using Lemma 4.7.5 with $h := |C| + \rho_1 + \rho_2$ and e as chosen above, we obtain an infinite set $P := \{p_1 < p_2 < p_3 < \ldots\} \subseteq \mathbb{N}$ with $p_1 > h$ and $p_j \equiv h \,(\mathrm{mod}\, e)$ for all $j \geqslant 1$, and mappings α_i taking the j-th element of D with respect to $<_i$ to p_j, for all $i \in \{1, 2\}$ and $j \in [1, n]$, such that

$$\mathfrak{M}_1 := (\mathbb{Z}, +, P, \alpha_1(\sigma^{\mathfrak{D}})) \equiv_{q''} (\mathbb{Z}, +, P, \alpha_2(\sigma^{\mathfrak{D}})) =: \mathfrak{M}_2,$$

for a number $q'' \in \mathbb{N}^+$ that will be fixed later on. We let q' be given by Lemma 4.7.5 for this choice of q''.

We define a first-order $((\sigma, P, +), (\sigma_{\Sigma,r}, +))$-interpretation $\mathcal{I}$ with the following properties, where $\mathcal{I}(\varphi)$ denotes the FO$[\tau]$-sentence constructed from φ according to the Interpretation Lemma:

(a) $\mathcal{I}(\mathfrak{M}_1)$ and $\mathcal{I}(\mathfrak{M}_2)$ are isomorphic to addition expansions of trees s_1 and s_2, respectively.

(b) s_1 and t are L-equivalent and s_2 and $\mathrm{hs}(t, u, v)$ are L-equivalent.

(c) We have $\mathrm{qr}(\mathcal{I}(\varphi)) \leqslant q''$, for a constant q'' which depends only on φ and not on t.

When we have obtained $\mathcal{I}$, we can finish the proof as follows:

$$\begin{aligned}
t \in L &\iff s_1 \in L && \text{(by (b))}\\
&\iff s_1 \models \varphi && (\varphi \text{ defines } L)\\
&\iff \mathcal{I}(\mathfrak{M}_1) \models \varphi && \text{(by (a) and addition-invariance)}\\
&\iff \mathfrak{M}_1 \models \mathcal{I}(\varphi) && \text{(Interpretation Lemma)}\\
&\iff \mathfrak{M}_2 \models \mathcal{I}(\varphi) && \text{(by } \mathfrak{M}_1 \equiv_{q''} \mathfrak{M}_2 \text{ and (c))}\\
&\iff \mathcal{I}(\mathfrak{M}_2) \models \varphi \iff s_2 \models \varphi \iff s_2 \in L \iff \mathrm{hs}(t, u, v) \in L.
\end{aligned}$$

It remains to define the interpretation $\mathcal{I}$.

Universe The universe of the interpreted structure is $[p_n]$. Here p_n can be identified by a first-order formula as the maximal element which belongs to P and where some unary predicate from σ is satisfied.

For the definition of the interpretation, we partition the interval $[p_n]$. Each part is used to simulate a certain part of the tree t. The first $c := |C|$ elements are used to simulate C, the following ρ_1 elements are used for the remainder block of t_1, and the next ρ_2 elements for the remainder block of t_2. This fixes the use of the first $h := c + \rho_1 + \rho_2$ elements.

The remaining elements of the interval $[h, p_n - 1]$ are partitioned into contiguous intervals of size e. Consider such an interval $I := [h + ie, h + (i+1)e - 1]$. If I is directly followed by a *P-element*, i.e. $h + (i + 1)e \in P$, we use its elements to simulate a subtree of t_1 or t_2 induced by one of the blocks of the structure $\mathfrak{D}$. We say that I is a *$\mathfrak{D}$-interval*. Recall that we have encoded the information about the successor and labelling relations of the blocks of t_1 and t_2 in $\mathfrak{D}$ and that the ordered structures $(\mathfrak{D}, <_1)$ and $(\mathfrak{D}, <_2)$ are embedded in the structures induced by P in $\mathfrak{M}_1$ and $\mathfrak{M}_2$, respectively.

Note that the previous arrangement assigns all parts of the trees $t = C[t_1, t_2]$ and $\mathrm{hs}(t, u, v) = C[t_2, t_1]$ to intervals. Indeed, it would be possible to recover the trees t and $\mathrm{hs}(t, u, v)$ from these parts of $\mathfrak{M}_1$ and $\mathfrak{M}_2$ by an interpretation. But our goal is to simulate structures which are addition expansions of tree structures and the restriction of the addition relation on $\mathbb{Z}$ to an arbitrary subset of $\mathbb{Z}$ is not necessarily an addition relation. But the restriction to an initial segment of the non-negative integers such as $[p_n]$ is an addition relation which we can copy to the simulated structures if we use this interval as their universe. Our description so far does not ascribe a meaning to an interval I as defined above if it is *not* directly followed by a P-element. In this case, we use I to simulate a copy of the idempotent context E and we say that I is an *E-interval.* All these additional copies of E will then be concatenated and inserted below the position where the original copy of E occured in t, i.e. the position which we have marked with the unary relation J_E. This way, in the place where $t = C[t_1, t_2]$ and $\mathrm{hs}(t, u, v) = C[t_2, t_1]$ contain the subtree $t_1 = DEt'_1$, the simulated trees will contain the subtree $DE^j t'_1$ for some number $j \in \mathbb{N}^+$ depending on p_n. The trees s_1 and s_2 simulated in $\mathfrak{M}_1$ and $\mathfrak{M}_2$ are L-equivalent to t and $\mathrm{hs}(t, u, v)$, respectively, since E is an idempotent context.

Now we continue the description of the interpretation in more detail. We define binary relations $\approx_C, \approx_{\mathrm{R}_1}, \approx_{\mathrm{R}_2}$ on the interval $[h]$ by

$$\begin{aligned} i \approx_C j &\iff i = j, \\ i \approx_{\mathrm{R}_1} j &\iff i = c + j, \\ i \approx_{\mathrm{R}_2} j &\iff i = c + \rho_1 + j. \end{aligned}$$

For instance, $i \approx_C j$ means that the element i is used to simulate the element $C@j$. The definition of $\approx_C$ in particular might seem somewhat superfluous, but it is used to unify the discussion below. Note that these relations are easily first-order definable in $\mathfrak{M}_1$ and $\mathfrak{M}_2$ since c, ρ_1 and ρ_2 are fixed.

We define similar relations for the $\mathfrak{D}$- and E-intervals. To this end, observe that an element i belongs to an $\mathfrak{D}$-interval in $\mathfrak{M}_1$ or $\mathfrak{M}_2$ if there exists an element $i' \in P$ with $i < i' \leqslant i + e$. If there is such an i' we say that i' is the *close P-successor* of i. We define the relations $\approx_{\mathfrak{D}}, \approx_E \subseteq [h, p_n - 1] \times [e]$ by

$$\begin{aligned} i \approx_{\mathfrak{D}} j &\iff i \equiv h + j \pmod{e} \text{ and } i \text{ has a close } P\text{-successor}, \\ i \approx_E j &\iff i \equiv h + j \pmod{e} \text{ and } i \text{ has no close } P\text{-successor}. \end{aligned}$$

That is, $i \approx_{\mathfrak{D}} j$ means that i plays the role of B$@j$ for some element B of $\mathfrak{D}$ and $i \approx_E j$ means that i plays the role of $E@j$ in a simulated copy of E. Note that these relations are easily first-order definable in $\mathfrak{M}_1$ and $\mathfrak{M}_2$ using addition since e and h are fixed.

Labels Consider a position $i \in [p_n]$. If i belongs to a $\mathfrak{D}$-interval, we use the unary relations of the embedded ordered copy of $\mathfrak{D}$ to determine the label of i. Otherwise, the label is determined solely based on the type of the interval of i and on the position of i inside this interval. For this the information about the fixed contexts C, E and the substructures of the tree structures of t_1 and t_2 which are induced by the remainder blocks R_1 and R_2 are "build into the interpretation". In summary, the unary predicate P_a, for $a \in \Sigma$, is satisfied at i if one of the following conditions holds:

- $i \approx_C j$ and $C@j$ is labelled by a in C.
- $i \approx_{R_k} j$ and $R_k@j$ is labelled by a in t_ℓ, for $\ell \in \{1, 2\}$.

- $i \approx_E j$ and $E@j$ is labelled by a in E.
- $i \approx_{\mathfrak{D}} j$ and $C_{j,a}$ is true at the close P-successor of i.

Successor relations Below we formulate defining conditions for the a-successor relation in the interpreted structure, for each $a \in \Pi$. That is, $\mathrm{S}_a(i_1, i_2)$ is true in the interpreted structure iff i_1, i_2 satisfy at least one of the conditions (i)-(viii) below, for each $i_1, i_2 \in [p_n]$.

For elements inside the intervals on the first h elements or in the same E-interval the conditions are straightforward.

(*i*) $i_1 \approx_A j_1$ and $i_2 \approx_A j_2$, and $A@j_2$ is the a-successor of $A@j_1$, for $A \in \{C, \mathrm{R}_1, \mathrm{R}_2\}$.

(*ii*) i_1 and i_2 belong to a common E-interval, $i_1 \approx_E j_1$, $i_2 \approx_E j_2$, and $E@j_2$ is the a-successor of $E@j_1$.

The definition of the remaining cases is complicated because we have to handle the insertion of the E-copies in the simulated tree. To this end, we have to do two things: (a) We have to fix some order in which the copies of E should be concatenated and (b) we have to simulate the insertion of the concatenated copies of E below the original copy.

Towards (a), we use the induced successor relation S_E of the restriction of the orderings on $\mathfrak{M}_1$ and $\mathfrak{M}_2$ to the least elements of the E-intervals. That is, S_E is defined by $\mathrm{S}_E(m_1, m_2)$ iff $m_1 {<} m_2$, $m_1 \approx_E 0$ and $m_2 \approx_E 0$ and $m \not\approx_E 0$ for all m with $m_1 {<} m {<} m_2$. Here m_1, m_2 are the *minimal elements* of their respective E-interval. We extend the relation S_E to all elements i_1 and i_2 which belong to E-intervals by setting $\mathrm{S}_E(i_1, i_2)$ iff $\mathrm{S}_E(m_1, m_2)$ is true for the minimal elements m_1 and m_2 of the E-intervals of i_1 and i_2.

Towards (b), recall that we have marked the block of t_1 which contains the parent of the hole of the copy of E in t_1 by the unary predicate J_E in $\mathfrak{D}$. We call this block B_E and we let $\ell_E \in [e]$ and $a_E \in E$ be such that the hole of E is the a_E-successor of its parent which is $E@\ell_E$.

Now we can formulate the remaining conditions.

(*iii*) The element i_2 belongs to a $\mathfrak{D}$-interval and its close P-successor p_ℓ is one of the first two P-elements, i.e. $\ell \in \{1, 2\}$, and $i_1 \approx_C j_1$ and $i_2 \approx_{\mathfrak{D}} 0$ for a number j_1 such that $C@j_1$ is the parent of the ℓ-th hole of C and the hole of C is the a-successor of $C@j_1$.

(It is crucial to make the following observation using the definitions of the linear orders $<_1$ and $<_2$ and the way $(\mathfrak{D}, <_1)$ and $(\mathfrak{D}, <_2)$ are embedded in $\mathfrak{M}_1$ and $\mathfrak{M}_2$: in the tree simulated in $\mathfrak{M}_1$, the previous condition inserts the root of the first block of t_1 at the first hole of C and the root of the first block of t_2 at the second hole of C, and the positions of these blocks are swapped in the tree simulated in $\mathfrak{M}_2$.)

(*iv*) The elements i_1 and i_2 belong to $\mathfrak{D}$-intervals with close P-successors p_{ℓ_1} and p_{ℓ_2}, respectively, and $i_1 \approx_{\mathfrak{D}} j_1$ and $i_2 \approx_{\mathfrak{D}} j_2$, and $S_{j_1, j_2, a}(p_{\ell_1}, p_{\ell_2})$ and either

- J_E is *not* true at p_{ℓ_1}, or

- J_E is true at p_{ℓ_1}, but $j_1 \neq \ell_E$, i.e. p_{ℓ_1} corresponds to the block B_E, but i_1 is not the position used to simulate the parent of the hole of the original copy of E in t_1.

(This way, we simulate the trees t_1 and t_2 using the relations of $\mathfrak{D}$ with the sole exception of the edge connecting the parent of the root of the original copy of E to E.)

(*v*) The element i_1 belongs to the unique $\mathfrak{D}$-interval whose close P-successor belongs to J_E, i_2 belongs to the first E-interval, $i_2 \approx_E 0$ (i.e. i_2 is the root of the first simulated copy of E), $i_1 \approx_{\mathfrak{D}} \ell_E$ and $a = a_E$.

(That is, we insert the first simulated copy of E at the hole of the original copy of E in t_1.)

(*vi*) The element i_2 belongs to a $\mathfrak{D}$-interval such that $S_{\ell_E, j_2, a_E}(p_{\ell_2}, p_{\ell_1})$ is true for a P-element p_{ℓ_1} with $J_E(p_{\ell_1})$, and $i_2 \approx_{\mathfrak{D}} j_2$, $a = a_E$, i_1 belongs to the last E-interval and $i_1 \approx_E \ell_E$, i.e. i_2 simulates the root of the subtree t_1' at the hole of E in t_1.

(That is, we insert t_1' below the last simulated copy of E.)

(*vii*) The elements i_1 and i_2 both belong to E-intervals, $\mathrm{S}_E(i_1, i_2)$ and $i_1 \approx_E \ell_E$, $i_2 \approx_E 0$ and $a = a_E$.

(That is, we concatenate the simulated copies of E according to the successor relation defined above.)

(*viii*) The element i_1 belongs to a $\mathfrak{D}$-interval whose close P-successor is an element p such that $K_{j_1, j_2, a}(p)$, $T_x(p)$, and $i_1 \approx_{\mathfrak{D}} j_1$, $i_2 \approx_{\mathrm{R}_x} j_2$, for some $x \in \{1, 2\}$.

(That is, we simulate the edges leading into the remainder block of t_x.)

This finishes the description of the intepretation. Note that, although the definition of $\mathcal{I}$ depends on E, C, R_1, R_2, the quantifier-rank of $\mathcal{I}(\varphi)$ can be bounded in terms of φ and independent of t. To see that this is true, recall that we have bounded the size of E, C, R_1, R_2 in terms of L only and hence there are only finitely many possible values for these parameters. For each choice of the parameters we obtain a different interpretation $\mathcal{I}$ and the maximum of these quantifier-ranks of $\mathcal{I}(\varphi)$ then yields the upper bound q''. $\square$

4.7.3 Closure under guarded vertical swaps

We continue with the proof of the following lemma.

Lemma 4.7.6. *Let L be a regular tree language. If L is definable by an* +-inv-FO*-sentence, then L is closed under strongly guarded vertical swaps.*

By Theorem 4.6.2, and since we have already proved closure under horizontal swaps in Lemma 4.7.2, this lemma establishes that addition-invariantly definable regular tree languages are closed under guarded swaps. It is possible to prove Lemma 4.7.6 by slight

Figure 4.7.1: A tree t (left) and the corresponding tree $\mathrm{split}(t,x)$ (right).

modification of the proof of Lemma 4.7.2, but we believe that it is more interesting to derive Lemma 4.7.6 by a reduction to Lemma 4.7.2 instead.

For each tree t of rank r and each node $x \neq \epsilon$ we define a tree $\mathrm{split}(t,x)$ of rank $r+1$ by disconnecting x from its parent and making it the $(r+1)$-successor of the root. See Figure 4.7.1. We extend the labelling by further information so as to make the map $(t,x) \mapsto \mathrm{split}(t,x)$ injective and its inverse a first-order interpretation.

Definition 4.7.7 ($\mathrm{split}(t,x)$)**.** Let $\Sigma' := \Sigma \cup (\Sigma \times \Pi_r)$. Let t be a tree of rank r and let $x \in \mathrm{dom}(t) \setminus \{\epsilon\}$. Suppose that x is the i-successor of a node y, for some $i \in \Pi_r$. Let u be the tree of rank $r+1$ obtained from the inner tree of the context $t[\epsilon, x)$ by changing the label of y from $t(y)$ to $(t(y), i)$ and by adding the node $r+1$ to the domain of this tree with an arbitrary label from Σ. Consider the context $C := (u, r{+}1)$ and let $s := t_{|x}$,

We define the tree

$$\mathrm{split}(t,x) \ := \ Cs.$$

We call trees which are obtained in this way *splits*. For each tree language L, we define the language of all splits of trees from L as

$$\mathrm{split}(L) \ := \ \{\mathrm{split}(t,x) : t \in L, x \in \mathrm{dom}(t).\}$$

We denote the language of all trees which are splits by

$$\mathrm{Splits} \ := \ \mathrm{split}(\mathcal{T})$$

For each tree that — like $\mathrm{split}(t,x)$ — contains a unique node whose label belongs to the alphabet $\Sigma \times \Pi_r$ we call that unique node the tree's *marked node*. The following crucial observation will be needed below.

Observation 4.7.8. *The map* $(t,x) \mapsto \mathrm{split}(t,x)$ *is injective.*

Proof. Let t, t' be trees of rank r and let $x \in \mathrm{dom}(t)$ and $x' \in \mathrm{dom}(t')$ such that $\mathrm{split}(t,x) = \mathrm{split}(t',x')$. We have to show that $(t,x) = (t',x)$. Let y and y' be the parents of x and x'. Let C, s and C', s' be such that $\mathrm{split}(t,x) = Cs$ and $\mathrm{split}(t',x') = C's'$ as in Definition 4.7.7. Since the node $r+1$ is the hole in both C and C', we have $s = s'$ and $C = C'$. Furthermore, y is the marked node of s. Its label is $(t(x), i) = (t'(x'), i)$ such that $x = yi = y'i = x'$. This implies that $t[\epsilon, x) = t'[\epsilon, x')$ and $x = x'$. Since $t = t[\epsilon, x) \cdot s$ and $t' = t'[\epsilon, x') \cdot s'$, we obtain $t = t'$. $\square$

As a next step, we show that relative to Splits, definability of L implies definability of $\mathrm{split}(L)$.

Lemma 4.7.9. *If L is addition-invariantly definable and regular, then there is an addition-invariantly definable regular language* $\mathrm{esplit}(L)$ *such that*

$$\mathrm{split}(L) = \mathrm{esplit}(L) \cap \mathrm{Splits}.$$

(Note that this does not imply the definability of $\mathrm{split}(L)$, since Splits is not FO-definable.)

Proof. Let $\Sigma' := \Sigma \times \Pi_r$. We show that the $(\sigma_{\Sigma',r+1}, \sigma_{\Sigma,r})$-interpretation $\mathcal{I}$ which reconstructs the Σ-labelled tree s of rank r from a Σ'-labelled tree $\mathrm{split}(s,x) \in \mathrm{Splits}$ of rank $r{+}1$ is first-order definable. That is, for each Σ'-labelled tree t of rank $r+1$ which belongs to Splits, $\mathcal{I}(t)$ is a Σ-labelled tree of rank r such that $\mathrm{split}(\mathcal{I}(t), x) = t$. The domain of $\mathcal{I}(t)$ is the same as the domain of t. All successor relations from t are copied to $\mathcal{I}(t)$ with the sole exception of the $(r+1)$-successor edge from the root. The node $r+1$ should become the node x in $I(t)$. Since $t \in \mathrm{Splits}$, there marked node y of t which is the only node labelled by a symbol $(a,i) \in \Sigma \times \Pi_r$. The interpretation makes x the i-successor of y.

It is obvious that this interpretation reverses the split-operation as desired.

We extend $\mathcal{I}$ to a $((\sigma_{\Sigma',r+1}, +), (\sigma_{\Sigma,r}, +))$-interpretation $\mathcal{I}'$ which simply copies the interpretation of the $+$-symbol from t to $\mathcal{I}(t)$. The sentence $\mathcal{I}'(\varphi)$ is addition-invariant. Suppose that there exist two distinct addition expansions $(t, +_1)$ and $(t, +_2)$ of a tree such that $(t, +_1) \models \tilde{\varphi}$ and $(t, +_2) \not\models \tilde{\varphi}$. Then $\mathcal{I}'(t, +_1) = (\mathcal{I}(t), +_1) \models \varphi$ but $\mathcal{I}'(t, +_2) = (\mathcal{I}(t), +_2) \not\models \varphi$. This contradicts the addition-invariance of φ.

We let $\mathrm{esplit}(L)$ be the language defined by $\mathcal{I}'(\varphi)$. By construction of $\mathcal{I}'$, it should be clear that $\mathrm{split}(L) = \mathrm{esplit}(L) \cap \mathrm{Splits}$. It remains to argue that $\mathrm{esplit}(L)$ is also regular. To this end, recall the classical result of Doner [Don70] and Thatcher and Wright [TW68] that a tree language is regular iff it is MSO-definable. Hence, L is MSO-definable by a sentence φ'. Then $\mathcal{I}(\varphi')$ is an MSO-sentence which defines the same language as $\mathcal{I}'(\varphi)$ and, consequently, $\mathrm{esplit}(L)$ is regular. $\square$

Note that the previous lemma relies crucially on the fact that the number of elements of t and $\mathrm{split}(t,x)$ is the same. Hence, the interpretation does not have to change the addition relation.

Proof of Lemma 4.7.6. Let L be an addition-invariantly definable regular tree language. Then $\mathrm{esplit}(L)$ is also regular and addition-invariantly definable by Lemma 4.7.9. Hence, by Lemma 4.7.2, $\mathrm{esplit}(L)$ is closed under k-guarded horizontal swaps for some $k \in \mathbb{N}$. We show that L is closed under strongly K-guarded vertical swaps for $K := 3(k+2)$.

Let $t \in L$ be a tree and let $t' := \mathrm{vs}(t, u_1, v_1, u_2, v_2)$ be a strongly K-guarded vertical swap of t. Let CC_1DC_2s be the standard decomposition of t. We have to show that $t' = CC_2DC_1s \in L$.

Let p_1 and p_2 denote the parents of v_1 and v_2. Let $v_1 = p_1 i$ for some $i \in \Pi$. Since v_1 and v_2 have isomorphic K-spheres in t, we obtain that $v_2 = p_2 i$ and that p_1 and p_2 have isomorphic $(K{-}1)$-spheres.

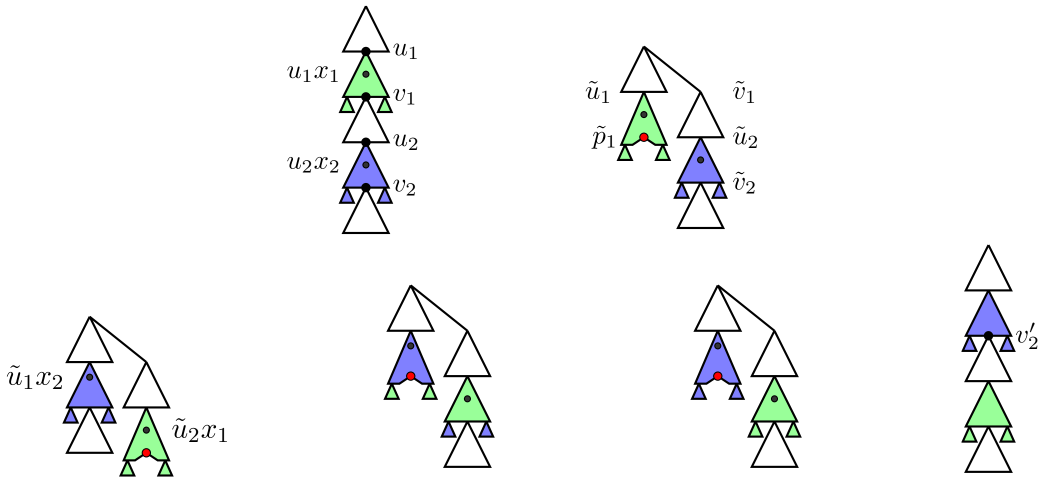


Figure 4.7.2: Trees from the proof of Lemma 4.7.6, in reading order: t, $\mathrm{split}(t, v_1)$, h, h', $h'_\alpha = \mathrm{split}(t', v_2')$, t'. The small trees are the subtrees at distance $k+1$ below x_1 and x_2, respectively. Note that the node v_2' has the same subtree in t' as the $(m+1)$-successor of the root of $\mathrm{split}(t', v_2')$, i.e. the right displayed subtree of the root of $\mathrm{split}(t', v_2')$.

We have $\mathrm{split}(t, v_1) \in \mathrm{split}(L) \subseteq \mathrm{esplit}(L)$, since $t \in L$. Our goal is to use the closure of $\mathrm{esplit}(L)$ under k-guarded horizontal swaps to show that $\mathrm{split}(t', v_2') \in \mathrm{esplit}(L)$, where v_2' is the node corresponding to the root of D in t'. Since $\mathrm{split}(t', v_2') \in \mathrm{Splits}$ and $\mathrm{esplit}(L) \cap \mathrm{Splits} = \mathrm{split}(L)$, we obtain $\mathrm{split}(t', v_2') \in \mathrm{split}(L)$. Then there must be some tree $t'' \in L$ and a node v_2'' such that $\mathrm{split}(t'', v_2'') = \mathrm{split}(t', v_2')$. According to Observation 4.7.8, we have $t'' = t'$ and hence $t' \in L$ and we are done.

First, we describe informally how we obtain $\mathrm{split}(t', v_2')$ from $\mathrm{split}(t, v_1)$ by k-guarded horizontal swaps and hence without leaving $\mathrm{esplit}(L)$. Recall that the node p_1 which corresponds to the parent of the hole of C_1 in t has a special label $(t(p_1), i)$ in $\mathrm{split}(t, v_1)$ which describes its relation to the node v_1 above which the split of t occurred. Note that the node p_2 which corresponds to the parent of the hole of C_2 in t' has the same label $(t'(p_2), i) = (t(p_1), i)$ in $\mathrm{split}(t', v_2')$. To obtain $\mathrm{split}(t', v_2')$ from $\mathrm{split}(t, v_1)$ we have do to the following things: (1) place C_2 (or, more precisely, the context corresponding to C_2 in $\mathrm{split}(t, v_1)$) below C and C_1 below D, (2) place s below C_1 and move the special label from the parent of the hole of C_1 to the parent of the hole of C_2. Step (1) can be done by a k-guarded horizontal swap between the nodes corresponding to u_1 and u_2, since these nodes are sufficiently far away from the nodes corresponding to p_1 and p_2 in $\mathrm{split}(t, v_1)$ and hence they are still k-similar. We cannot immediately realise step (2) in the same way by a k-guarded horizontal swap, since the nodes which should become the parents of the roots of D and s, i.e. the nodes p_1' and p_2' corresponding to p_1 and p_2 in the tree h obtained after step (1), are not k-similar. But, the $(K-1)$-spheres of p_1 and p_2 in t are isomorphic and, by strong K-guardedness, p_1 and p_2 are sufficiently far way from u_1 and u_2. Hence, we can ascend for $(k+1)$-steps on the paths to p_1 and p_2 to find k-similar nodes x_1 and x_2. We can then perform a k-guarded horizontal swap between the corresponding nodes x'_1 and x'_2 of h to obtain a tree h' which almost achieves step (2): we have moved the special label and s to the right position, but the subtrees of C_1 and C_2 below the other nodes at distance $k+1$ from x'_1 and x'_2 are now in the wrong place; note that the trees "between" x'_1 and x'_2 and these nodes look the same in C_1 and C_2. We can revert this mistake, by exchanging the subtree of each node at distance exactly $k+1$ below p'_1, except p'_1, with the corresponding node below p'_2 by k-guarded horizontal swaps. This way, we complete step (2) and it is straightforward to see that the obtained tree is equal to $\mathrm{split}(t', v'_2)$.

For each node $w \in \mathrm{dom}(t)$, let $\tilde{w}$ be the corresponding node of $\mathrm{split}(t, v_1)$, i.e. $\tilde{w} := (r+1)x$ if $w = v_1 x$ and $\tilde{w} := w$, otherwise. We note some facts about the relation between the types of nodes w in t and $\tilde{w}$ in $\mathrm{split}(t, v_1)$.

1. $\mathrm{split}(t, v_1)_{|\tilde{w}} = t_{|w}$, if $w \parallel p_1$ or $w \trianglerighteq v_1$.
2. $\mathrm{split}(t, v_1)^k_{|\tilde{w}} = t^k_{|w}$, if $w \trianglelefteq p_1$ and $\mathrm{dist}(w, p_1) > k$.

By strong K-guardedness, we have $\mathrm{dist}(u_1, v_1) \geqslant 2K+1$ and $\mathrm{dist}(u_2, v_2) \geqslant 2K+1$. Let u_1x_1 and u_2x_2 be the nodes with $u_1 \lhd u_1x_1 \lhd p_1$ and $u_2 \lhd u_2x_2 \lhd p_2$ and $\mathrm{dist}(p_1, u_1x_1) = \mathrm{dist}(p_2, u_2x_2) = k+1$. Since p_1 and p_2 have isomorphic $(3k+2)$-spheres, we obtain that $t^{2k+1}_{|u_1x_1} = t^{2k+1}_{|u_2x_2}$. Let $u_1x_1y_1, \ldots, u_1x_1y_\alpha$ be an enumeration of the nodes at distance $k+1$ below u_1x_1 in t where $u_1x_1y_1 = p_1$. By $(2k+1)$-similarity of u_1x_1 and u_2x_2, we obtain that $u_2x_1y_1, \ldots, u_2x_2y_\alpha$ is an enumeration of the nodes at distance $k+1$ below u_2x_2 in t where $u_2x_2y_1 = p_2$ and such that $u_1x_1y_i$ is k-similar to $u_2x_2y_i$, for each $i \in [2, \alpha]$.

Let $h := \mathrm{hs}(\mathrm{split}(t, v_1), \tilde{u}_1, \tilde{u}_2)$ and note that this is a k-guarded swap. Now let $h' := \mathrm{hs}(h, \tilde{u}_2x_1, \tilde{u}_1x_2)$. Since the first swap was k-guarded, we know that $\tilde{u}_2x_1$ and $\tilde{u}_1x_2$ are k-similar in h and hence the second swap is k-guarded. Now we swap all nodes at distance $k+1$ below $\tilde{u}_2x_1$ and $\tilde{u}_1x_2$ except the nodes corresponding to p_1 and p_2. To this end, we consider the sequence $h'_2 := \mathrm{hs}(h', \tilde{u}_1x_2y_2, \tilde{u}_2x_1y_2)$, $h'_3 := \mathrm{hs}(h'_2, \tilde{u}_1x_2y_3, \tilde{u}_2x_1y_3)$, and so on till h'_α. Note that all these swaps are k-guarded. From the discussion above, it is clear that $h'_\alpha = \mathrm{split}(t', v'_2)$. See also Figure 4.7.2. $\square$

4.7.4 Closure under transfer

We show that each regular tree language definable by an $+$-inv-FO-sentence is closed under transfer. We can build on the main technical lemma from the proof of the corresponding result for word languages [SS10]. Recall that the canonical addition expansion of a word w is the structure $(w, +)$ where $+$ is the addition relation induced by the natural linear order on w. For a given alphabet Γ and languages $L_1, L_2 \subseteq \Gamma^*$, we say that an $\mathrm{FO}[\sigma_{\Gamma,1}, +]$-sentence φ *separates* L_2 from L_1 if the canonical addition expansion of each word from L_1 satisfies φ and if no canonical addition expansion of a word from L_2 satisfies φ.

Lemma 4.7.10 (Proposition 3.3 of [SS10]). *Let* $n \in \mathbb{N}$ *with* $n \geqslant 2$, $y \in \Sigma^*$, $\bar{x} \in (\Sigma \times \{1\})^*$ *and* $\bar{z} \in (\Sigma \times \{2\})^*$. *Let* $\Gamma := \Sigma \cup (\Sigma \times \{1, 2\})$. *For all* $a, b \in \mathbb{N}$, *let*

$$L_{n,a,b} := \{w \in y\bar{x}(\bar{x}\bar{z}|\bar{z}\bar{z})^* : |w|_{\bar{x}}, |w|_{\bar{z}} \geqslant n, |w|_{\bar{x}} \equiv a\,(mod\,n), |w|_{\bar{z}} \equiv b\,(mod\,n)\}.$$

There exists no $\mathrm{FO}[+, \sigma_{\Gamma,1}]$*-sentence that separates* $L_{n,1,0}$ *from* $L_{n,0,1}$.

Here $|w|_{\bar{x}}$ and $|w|_{\bar{z}}$ denote the number of $\bar{x}$ and the number of $\bar{z}$ factors in the given factorisation of w.

We proceed with a proof by contradiction: If a regular tree language L is not closed under transfer, we show by an interpretation argument that there exists an FO-sentence separating the word language $L_{n,1,0}$ from the language $L_{n,0,1}$ for $n := \omega_L$. This is akin to what is done in [SS10] to prove that every regular *word* language is closed under transfer, the difference being that we have to simulate trees in words.

Lemma 4.7.11. *Let* L *be a regular tree language. If* L *is definable by an* $+$-inv-FO*-sentence, then* L *is closed under transfer.*

Proof. Assume that L is not closed under transfer. This means, there exist contexts C_1, C_2 with $|C_1| = |C_2|$ and a 2-template T, such that the following holds:

$$T\langle C_1^{\omega+1}, C_2^{\omega}\rangle \in L \quad \text{and} \quad T\langle C_1^{\omega}, C_2^{\omega+1}\rangle \notin L.$$

Because C_1^{ω} and C_2^{ω} are idempotent, we may repeat both contexts in the given trees without affecting membership in L. Therefore, the following holds for the trees $t_{i,j} := T\langle C_1^i, C_2^j\rangle$, for all $i, j \in \mathbb{N}^+$:

$$i \equiv 1 \,(\mathrm{mod}\, \omega), j \equiv 0 \,(\mathrm{mod}\, \omega) \implies t_{i,j} \in L, \tag{4.12}$$
$$i \equiv 0 \,(\mathrm{mod}\, \omega), j \equiv 1 \,(\mathrm{mod}\, \omega) \implies t_{i,j} \notin L. \tag{4.13}$$

As we are aiming at a contradiction to Lemma 4.7.10, we fix the parameters of the lemma. To each tree s we assign the word $s(s@0)\, s(s@1) \cdots s(s@|s|-1)$, i.e. the word of its labels ordered according to the lexicographic order on s. Let y be the word obtained from the underlying tree of the template T in that way. Let x and z be the words obtained from the inner tree of the contexts C_1 and C_2, respectively. Let $\bar{x}$ be the word x with each symbol a that occurs in x replaced by the tuple $(a, 1)$, and let $\bar{z}$ be obtained from z by replacing each symbol a of z by $(a, 2)$. We say that positions with a label $(a, 1)$ or $(a, 2)$ are *tagged* by 1 or 2, respectively. Let $\sigma := (\sigma_{\Sigma,r}, +)$ and $\gamma := (\sigma_{\Gamma,1}, +)$, where r is the fixed bound on the rank of trees. We define a first-order (γ, σ)-interpretation $\mathcal{I}$ such that, for each word $w \in \mathcal{L}(y\bar{x}(\bar{x}\bar{z}|\bar{z}\bar{z})^*)$, the structure $\mathcal{I}(w, +)$ is isomorphic to an addition expansion of a tree $t_{i,j}$ such that $i = |w|_{\bar{x}}$ and $j = |w|_{\bar{z}}$.

Before we define the interpretation, we show how we use it to finish the proof. Let φ be the $+$-inv-FO-sentence defining L and let $\mathcal{I}(\varphi)$ be the sentence of the Interpretation Lemma. Let $n := \omega$ and consider the languages $L_{n,1,0}$ and $L_{n,0,1}$ of Lemma 4.7.10. Let $w \in \mathcal{L}(y\bar{x}(\bar{x}\bar{z}|\bar{z}\bar{z})^*)$ and let $i := |w|_{\bar{x}}$ and $j := |w|_{\bar{z}}$. Then $\mathcal{I}(w, +)$ is isomorphic to some addition expansion of $t_{i,j}$.

If $w \in L_{n,1,0}$, then $i \equiv 1 \,(\mathrm{mod}\, \omega)$ and $j \equiv 0 \,(\mathrm{mod}\, \omega)$. Thus $t_{i,j} \in L$ by (4.12). Since φ is addition-invariant and defines L, we have $\mathcal{I}(w, +) \models \varphi$. Then $(w, +) \models \mathcal{I}(\varphi)$ by the Intepretation Lemma.

On the other hand, if $w \in L_{n,0,1}$, then $i \equiv 0 \,(\mathrm{mod}\, \omega)$ and $j \equiv 1 \,(\mathrm{mod}\, \omega)$. Thus $t_{i,j} \notin L$ by (4.13). Since φ is addition-invariant and defines L, we have $\mathcal{I}(w,+) \not\models \varphi$. Then $(w,+) \not\models \mathcal{I}(\varphi)$ by the Interpretation Lemma.

Hence, $\mathcal{I}(\varphi)$ separates $L_{n,0,1}$ from $L_{n,1,0}$ which contradicts Lemma 4.7.10 and finishes the proof.

It remains to define the interpretation $\mathcal{I}$. Consider a word $w \in \mathcal{L}(y\bar{x}(\bar{x}\bar{z}|\bar{z}\bar{z})^*)$ of length n and its canonical addition expansion $(w,+)$. The structure $\mathcal{I}(w,+)$ has the same universe as w, i.e. the set $[n]$. The addition relation is copied from w. We define the label and successor relations in such a way that on the positions belonging to the prefix y the interpretation simulates the nodes of the template T, on each $\bar{x}$-factor it simulates a copy of C_1, on each $\bar{z}$-factor it simulates a copy of C_2, and all copies of C_1 are concatenated and inserted at the first insertion point of T and the same happens for C_2 and the second insertion point of T. To this end, the unary label relations can just be copied from the word, removing the tags 1 and 2, i.e. $\mathrm{P}_a^{\mathcal{I}((w,+))} = \mathrm{P}_a^{(w,+)} \cup \mathrm{P}_{(a,1)}^{(w,+)} \cup \mathrm{P}_{(a,2)}^{(w,+)}$ for each $a \in \Sigma$. Let $u_1 := \bar{x}$ and $u_2 := \bar{z}$.

Then the a-successor relation S_a is defined as follows, for each $a \in \Pi$.

- On the elements of y, the successor relations are defined as in T with the exception that we insert the position where we simulate the root of the first copy of C_i at the i-th point of T, for each $i \in \{1,2\}$. That is, if the i-th point of T is the a-successor of some node v such that $v = T@k$ then the a-successor of the position k in $\mathcal{I}((w,+))$ is the first position of the first occurrence of u_i with respect to $<^{(w,+)}$. This position can be identified with the help of the tags 1 and 2.

- Between elements of the same u_i-factor, the successor relations are defined according to C_i.

- The simulated copies of C_1 and C_2 are linked to the remaining tree as follows. Suppose that the hole of C_i is the a_i-successor of a node p_i and let k_i be such that $p_i = C_i@k_i$.
  - The a_i-successor of the k_i-th element of some occurrence of u_i which is not the last occurrence of u_i is the first position of the next occurrence of u_i.
  - The a_i-successor of the k_i-th element of the last occurrence of u_i is the position of y which corresponds to the i-th point of T.

All of the above conditions can be easily expressed in $\mathrm{FO}[\sigma_{\Gamma,1},+]$. For this, we use the tags 1 and 2 to distinguish between factors which should be treated as u_1 and u_2 and the fact that the contexts C_1 and C_2 and the template T are fixed. □

From Lemma 4.7.11, Lemma 4.7.2, and Lemma 4.7.6, we immediately obtain Theorem 4.7.1.

4.8 Conclusion

In this chapter, we have obtained a decidable characterisation of the regular tree languages which are definable by first-order sentences with cardinality predicates. We have used this characterisation to prove that these languages coincide with regular tree languages which are addition-invariantly definable.

Several questions remain open. As we have pointed out, an elementary upper bound for the time complexity of deciding closure under transfer can be obtained. It would be interesting understand the complexity of this problem better.

We have shown that for each addition-invariantly definable sentence φ which defines a regular tree language there is a first-order sentence $\tilde{\varphi}$ which defines the same tree language using cardinality predicates but no addition. It is not clear from our proofs how the size of $\tilde{\varphi}$ grows with the size of φ. Can this growth be bounded by an elementary function?

Do there exist addition-invariantly definable tree languages which are not regular? This question has been raised in [SS10] for word languages and remains open. For some particular classes of languages such as the bounded, the commutative, and the deterministic context free languages it was shown in [SS10] that addition-invariant definability implies regularity. Hence, restricted to such languages, addition-invariant definability is also equivalent to $\mathrm{FO}_{\mathrm{card}}$-definability.

Logic on classes of structures of bounded tree-depth

In this section we consider FO, FO+MOD, and MSO and the order-invariant formulae of this logics on classes of structures of *bounded tree-depth*. We relate the expressive power of these logics to each other and consider questions about the relative *succinctness* of these logics.

5.1 Introduction

We start with a introduction where we give a short overview over the notion of tree-depth and its role in logic, then we review some results about order-invariant FO and MSO, and we explain shortly what we mean by succinctness. After that, the contributions of this chapter are summarised.

5.1.1 The notion of tree-depth

The notion of *tree-depth* was introduced by Nešetřil and de Mendez in [NdM06]. There the tree-depth of a connected graph G was defined as the minimum height of a rooted tree T such that G is a subgraph of the *closure* $\operatorname{clos}(T)$ of T, i.e. the underlying undirected graph of the transitive closure of T.[1] See Figure 5.1.1 for an example which is adapted from [NdM12]. Under different names (e.g. height of *elimination trees*), this and similar

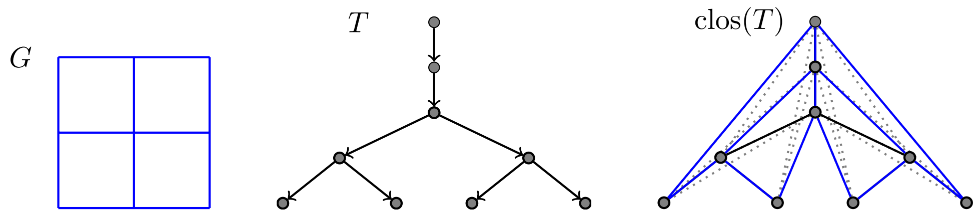


Figure 5.1.1: A 3×3-grid and a tree T whose closure contains G.

[1] We do not need a more precise statement of this definition here; the well-known characterisation of tree-depth that is used in this chapter is introduced in Section 5.3.2.

notions have been studied much earlier. See [NdM12] which contains an overview of various properties of tree-depth of graphs. The notion of tree-depth plays a central role in the study of sparse graph classes, in particular *nowhere dense* graph classes and graph classes of *bounded expansion*, that was initiated by Nešetřil and de Mendez and laid out in their book [NdM12].[2] First-order logic has many desirable properties on nowhere dense graph classes. For example, first-order definable properties can be decided in time $O(n^{1+\epsilon})$, for every $\epsilon > 0$ (cf. [GKS14]), and the *homomorphism preservation theorem*[3] holds restricted to nowhere dense graph classes which are closed under substructures and disjoint unions [Daw10, NdM10]. Tree-depth has also proved to be useful in applications to logic which are not immediately related to sparse graphs. For example, it plays an important role in Rossman's proof that the homomorphism preservation theorem holds for the class of all finite structures [Ros08].

Here we are interested in classes of graphs (and structures) of *bounded tree-depth*. That is, in classes for which there is a constant c such that the tree-depth of each graph in the class is at most c. Various works have considered different aspects of the algorithmic properties of MSO on graphs of bounded tree-depth (e.g. [EJT12],[GH15],[Lam14]). A remarkable result of a more structural nature was obtained in [EGT12]. While it is evident that MSO is more expressive than FO on the class of all finite graphs — as witnessed by the reachability query which can be expressed in MSO but not in FO — the same is not necessarily true for other, restricted classes of structures. A simple example of a class $\mathcal{C}$ where MSO and FO have the same expressive power, written MSO $=_{\mathcal{C}}$ FO, is the class $\mathcal{C}$ of all finite sets, viewed as structures over the empty signature.[4] For which other classes can this happen? The following theorem answers two important instances of the question for MSO and MSO_2, where MSO_2 is the extension of MSO_2 with the capability to quantify over subsets of edges in addition to subsets of vertices.

Theorem 5.1.1 ([EGT12]). *Let $\mathcal{C}$ be a class of graphs.*

1. *If $\mathcal{C}$ is closed under subgraphs, then* MSO $=_{\mathcal{C}}$ FO *iff $\mathcal{C}$ has bounded tree-depth.*
2. *If $\mathcal{C}$ is closed under induced subgraphs, then* $\mathrm{MSO}_2 =_{\mathcal{C}}$ FO *iff $\mathcal{C}$ has bounded tree-depth.*

Note that characterising the classes which are closed under induced subgraphs and for which MSO $=_{\mathcal{C}}$ FO is an open problem (cf. [EGT12], [EGT16]).

A part of the proof of Theorem 5.1.1 consists in the proof of the following theorem.

Theorem 5.1.2 ([EGT12]). *For each $d \in \mathbb{N}$, each class $\mathcal{C}$ of coloured graphs of tree-depth at most d, and each* MSO*-sentence φ there is an* FO*-sentence $\tilde{\varphi}$ which is equivalent to φ on $\mathcal{C}$.*

Note that this theorem does not tell us anything about the relationship between the size of φ and the size of $\tilde{\varphi}$.

[2]Shorter surveys on these topics, covering the connection to tree-depth, can also be found in [NdM15], [GKS13].

[3]That is, for such classes $\mathcal{C}$, a first-order sentence is preserved under homomorphisms on $\mathcal{C}$ iff it is equivalent to an existential sentence

[4]This follows immediately, for instance, from [Lib04, Proposition 7.12].

5.1.2 Succinctness

One of the major goals of finite model theory is surely the understanding of the expressive power of different logics on finite structures which is very often tied to questions from complexity theory. But even if we know that two logics L_1 and L_2 have the same expressive power, it could happen that some properties can be expressed in L_1 much more *succinctly* than in L_2, i.e. by shorter formulae. There are different ways to interpret a result of this kind. On the one hand, this could be seen as showing that L_2 is less suited for the use in applications, e.g. in formal verification or databases, since the formulae expressing desirable properties in L_2 might become too complicated. On the other hand, the algorithmic properties of L_2 could be much better than the algorithmic properties of L_1 and hence the use of L_2 could still be preferable. The fact that L_1 is more succinct than L_2 can be seen as a kind of explanation for this. Whatever interpretation one prefers, understanding succinctness questions seems helpful to assess the potential for use of a logic in applied settings. As an example where the latter situation arises, consider the paper [LSW01] which introduced a modal logic M which has the same expressive power as the two-variable fragment of FO (FO^2) on finite structures. The satisfiability problem for M is shown to be EXP-complete and the satisfiability problem for FO^2 is known to be NEXP-complete. As discussed in the paper, this implies that FO^2 must be more succinct than M, unless EXP = NEXP.

We consider another very simple concrete example of the kind of questions where understanding succinctness questions is helpful. As a consequence of Theorem 5.1.2, for each class $\mathcal{C}$ of bounded tree-depth graphs and each fixed MSO-sentence φ there is a polynomial time algorithm which decides the class defined by φ on $\mathcal{C}$. [5] For this, we take an FO-formula $\tilde{\varphi}$ which is equivalent to φ on $\mathcal{C}$ and which exists according to Theorem 5.1.1. Instead of evaluating φ to check if a graph from $\mathcal{C}$ is a model of φ, we evaluate $\tilde{\varphi}$. This can be done in polynomial time using a naïve algorithm for evaluating FO-formulae which follows from the definition of the semantics of FO. The size of $\tilde{\varphi}$ occurs as a hidden constant in the running time of this algorithm and understanding the size of $\tilde{\varphi}$ in terms of the size of φ helps us to understand this important detail.[6]

5.1.3 Order-invariant FO and MSO

While the definition of order-invariance occurs very naturally in database theory and descriptive complexity theory (see Chapter 1), the implications of this definition are not clear at all. At first sight, for FO, it is not even clear if allowing the invariant use of a linear order adds to the expressive power. In contrast, it is easy to see that there is a sentence φ_{even} of order-invariant MSO (<-inv-MSO) which states that the size of the universe of a structure is even.[7] It is well-known that this cannot be expressed by an MSO-

[5] Of course, this follows also from the well-known theorem of Courcelle which establishes this result for more general classes of graphs of bounded tree-width (cf. e.g. [CE12]).

[6] For this example, i.e. evaluating MSO-formulae on bounded tree-depth graphs, the constants in the complexity have been investigated by other methods in [GH15].

[7] This formula states that there exist two sets which partition the universe such that the least element in the order belongs to the first set, the maximum element belongs to the second set, and an element belongs

sentence (cf. [Lib04, Proposition 7.12]). Gurevich has given an example (cf. e.g. [EF99], [Lib04]) of an $<$-inv-FO-definable property of finite structures which is not FO-definable. This example uses *powerset structures*, i.e. structures of the form $(2^X, \subseteq)$ where X is a set and $\subseteq$ is the subset relation on X. On such structures, $<$-inv-FO can simulate $<$-inv-MSO on the underlying set X which corresponds to the singleton sets of 2^X. The $<$-inv-MSO-sentence φ_{even} then yields an $<$-inv-FO-sentence which expresses that the underlying set of a powerset structure has even cardinality. It can be shown that FO cannot do this. There are other known examples (cf. [Sch13] for a survey). It is natural to ask if there are any non-trivial limitations to the gain in expressive power that can be achieved by the invariant use of the order. The most general result to this extend was obtained by [GS00] who showed that, like FO-formulae, $<$-inv-FO-formulae are *Gaifman local*. This restricts the expressive power of order-invariant formulae severely. Even more ambitiously, since order-invariance is undecidable (cf. [Gur88]), it can be asked if there is a logic with a decidable syntax which has exactly the same expressive power as $<$-inv-FO. Somewhat less ambitiously than this, one could ask for a logic L with a decidable syntax which constitutes and upper bound for the expressive power of $<$-inv-FO, i.e. every $<$-inv-FO-definable class of finite structures is also L-definable, and which is as weak as possible. It is known (cf. [Ott00, Corollary 8]) that the infinitary bounded-variable logic with counting quantifiers $\mathrm{C}^{\omega}_{\infty\omega}$ cannot serve the role of L, i.e. there are $<$-inv-FO-definable classes of finite structure which are not $\mathrm{C}^{\omega}_{\infty\omega}$-definable. This also rules out fixed point logics with and without counting and extensions of FO by arbitrary monadic generalised quantifiers. It is an open problem if MSO could serve the role of L (cf. [BS09b]).

For the class of all finite structures, these questions seem to be very hard to answer. However, the structures employed in the examples which show that $<$-inv-FO is more expressive than FO are *dense*, i.e. their Gaifman graphs contain large cliques. The study of the expressive power of $<$-inv-FO on particular classes of sparse structures has been initiated by Benedikt and Segoufin [BS09b]. They showed that on various kinds of trees (viewed as successor structures without the descendant relation) $<$-inv-FO is no more expressive than FO. The proofs rely on an algebraic characterisation of FO-definability which was obtained by the same authors in [BS09a] (the same characterisation which we also used in Section 4.4) and on the Gaifman-locality result of [GS00]. Similar results have been obtained by Niemistö [Nie07] who also considered the extension of $<$-inv-FO by modulo-counting quantifiers. Apart from these results, not much seems to be known about the expressive power of order-invariant FO on sparse graphs. Benedikt and Segoufin [BS09a] also considered the question which logics (without numerical predicates) constitute an upper bound on the expressive power of $<$-inv-FO on sparse graphs. They showed that $<$-inv-FO is contained in MSO on all graphs of bounded tree-width. Very recently, a much more general result was published by Elberfeld, Frickenschmidt, and Grohe [EFG16]. They show that $<$-inv-FO is contained in MSO on graphs classes which exclude a forbidden minor. This includes, for instance, classes of bounded tree-width and planar graphs.

---

to the first set iff its successor belongs to the second set.

Order-invariant MSO ($<$-inv-MSO) has also received some attention in the literature. In [Cou96], Courcelle studied the expressive power of $<$-inv-MSO. He showed that on forests, $<$-inv-MSO and the extension of MSO by first-order modulo-counting quantifiers (CMSO) have the same expressive power. The same idea as for the sentence φ_{even} mentioned above shows that, on ordered structures, all modulo-counting quantifiers can be defined in MSO. Hence, CMSO is contained in $<$-inv-MSO. Courcelle [Cou96] asked if this containment is strict. This was answered affirmatively by Ganzow and Rubin [GR08] who gave an example of an $<$-inv-MSO-definable class of finite structures which is not CMSO-definable. Recently, it has been shown that CMSO and $<$-inv-MSO have the same expressive power on graphs of bounded tree-width [EFG16, BP16].

5.2 Contributions

We discuss the contributions of this chapter and their relation to other results from the literature. The main results of this chapter have been announced in the conference paper [EEH14]. A manuscript containing an extended version of this paper has been made available on the Internet [EEH16] and has been submitted to a journal. The following chapter is based on this manuscript.

Undecidability

The *first main result* of this chapter shows that $<$-inv-FO *is undecidable* on the class of all finite structures of tree-depth at least 2 (Theorem 5.4.2). This serves as a motivation to look for an effective syntax for $<$-inv-FO on structures of bounded tree-depth.

Succinctness of $<$-inv-FO and FO

Our *second main result shows* that, on classes of structures of tree-depth at most d, each $<$-inv-FO-sentence is equivalent to an FO-sentence whose size is at most d-fold exponential in the quantifier-rank of the $<$-inv-FO-sentence (Theorem 5.4.1). We also show that $<$-inv-FO is non-elementarily more succinct than FO on finite structures (Theorem 5.7.5) and we conclude from this that the height of the tower of exponentials in the previous result cannot be independent of d (Corollary 5.7.9).

Note that there are several possible ways to derive the mere result that $<$-inv-FO has the same expressive power as FO on bounded tree-depth *graphs* from known results. The result follows immediately from the fact that $<$-inv-FO is contained in MSO on bounded tree-width structures [BS09b] and Theorem 5.1.2.[8] Another approach for a proof of the

[8] Note that the status of the theorem which shows that $<$-inv-FO is contained in MSO on bounded tree-width structures was somewhat unclear at the time when our results were obtained. Its proof relies on the possibility to define the tree-decompositions of graphs of bounded tree-width in the graphs using MSO-formulae. The correctness of the published proof sketches of this result have been doubted by several prominent researchers. The situation has changed recently. Bojanczyk and Pilipczuk [BP16] showed that tree-decompositions can indeed be defined by MSO-formulae. Furthermore, Elberfeld, Frickenschmidt, and Grohe [EFG16], have provided a new proof that $<$-inv-FO is contained in MSO on bounded tree-width

expressivity result for graphs uses the result of [BS09b] that $<$-inv-FO and FO have the same expressive power on trees. By proving that tree-decompositions of bounded tree-depth graphs can be defined in the graphs using FO-formulae, the result could also be proved.[9]

Neither of these methods seems well-suited to prove the succinctness results that we obtain. In contrast, our approach relies only on methods from mathematical logic such as Ehrenfeucht-Fraïssé-games, types, and compositions methods. It is a remarkable fact that these methods can be used to prove results about the expressive power of $<$-inv-FO. The results of Benedikt and Segoufin [BS09b] relied on automata theoretic and algebraic methods. More recently, the applicability of type and composition methods for obtaining results about order-invariant formulae has also been investigated in the paper [BL16]. This paper studies order-invariant types whereas our methods consider the types of particular orderings of structures.

$<$-inv-MSO and FO+MOD

Using similar methods as for the second main result, we prove our *third main result* that $<$-inv-MSO *has the same expressive power as* FO+MOD on bounded tree-depth structures (Theorem 5.5.1). Here, we do not study the corresponding succinctness questions since it follows from known results of [GS05] that $<$-inv-MSO is already non-elementarily more succinct than FO+MOD on structures over the empty signature.

To the best of our knowledge, this expressivity result does not follow from any known results. While we know that $<$-inv-MSO and CMSO have the same expressive power on bounded tree-depth structures [BP16], it was not known that CMSO and FO+MOD have the same expressive power on such structures. Since CMSO is contained in $<$-inv-MSO, our third main result also establishes this result. This can be seen as an answer to a question from the conclusion of [EGT12]. The authors asked which extension of first-order logic corresponds to CMSO on bounded tree-depth graphs.

Succinctness of MSO and FO

Our *fourth main result* considers the succinctness of MSO and FO on bounded tree-depth structures. We show that, on structures of tree-depth at most d, each MSO-sentence is equivalent to an FO-sentence whose size is at most d-fold exponential in the quantifier rank of the MSO-sentence (Theorem 5.6.1). For this, we use similar proof methods as for our second and third main result.

This result can be seen as a quantitative refinement of Theorem 5.1.2. Whereas the proof of Theorem 5.1.2 in [EGT12] relies on an involved constructive Feferman-Vaught-like theorem, our approach is simpler since it needs only a non-constructive folklore composition lemma which has a trivial game-based proof. This simplification benefits the analysis of the formula size in our proof.

---

structures which does not rely on the possibility to define tree-decompositions.

[9] The manuscript [EEH16] contains a construction of an FO-formula which, for some fixed tree-depth d, defines a tree-decomposition in each graph of tree-depth at most d.

$\varphi \in$	$<$-inv-FO	MSO	$<$-inv-MSO
$\psi \in$	FO	FO	FO+MOD
$\lvert\psi\rvert$	$\mathrm{EXP}_d(q)$	$\mathrm{EXP}_d(q)$	non-elementary
$\mathrm{qad}(\psi)$	$3d$	$3d$	$3d$

Figure 5.2.1: Summary of the results about the expressive power and succinctness of the considered logics. A formula φ of quantifier rank q is translated into a formula ψ that is equivalent to φ on structures of tree-depth at most d.

We also prove a lower bound which shows that the dependency of the FO-formula size on the tree-depth in this result is essentially optimal (Theorem 5.7.1).

In the succinctness upper bounds, we also consider the quantifier alternation depth of the constructed FO-sentence. In all cases, we show that the alternation depth is bounded by a linear function of the tree-depth and hence that the quantifier alternation hierarchy considered by Chandra and Harel [CH82] collapses on bounded tree-depth structures. This complements a result of [CH82] which showed that the quantifier alternation hierarchy for FO is strict on trees of unbounded height. Figure 5.2.1 summarises our succinctness results.

5.3 Preliminaries

We introduce definitions and terms which will be used throughout this chapter.

Growth of functions The *d-fold exponential function* $\exp_d(n)$ is inductively defined by $\exp_0(n) := n$, and $\exp_{(d+1)}(n) := 2^{\exp_d(n)}$. We let $\mathrm{tower}(d) := \exp_d(1)$. The *class of functions that grow at most d-fold exponentially* is

$$\mathrm{EXP}_d \;:=\; \{f : \mathbb{N} \to \mathbb{N} : \text{ there is a } c \in \mathbb{N} \text{ such that } f(n) \leqslant \exp_d(n^c) \text{ for each } n \in \mathbb{N}\}.$$

We say that a function $f : \mathbb{N} \to \mathbb{N}$ is *elementary* if $f \in \mathrm{EXP}_d$, for some $d \in \mathbb{N}$.

Linear orders For two linear orders $\preceq_1$ and $\preceq_2$ on disjoint sets M_1 and M_2, we define a linear order $\preceq_1 + \preceq_2$ on $M_1 \cup M_2$, the *(ordered) sum* of $\preceq_1$ and $\preceq_2$, as $\preceq_1 \cup \preceq_2 \cup (M_1 \times M_2)$.

Proviso 7 (Order relations are linear)**.** *Throughout this chapter, we consider only order relations which are linear and hence, when we say "order", we mean (strict) "linear order".*

An *ordered structure* is a structure where the binary relation symbol $<$ is interpreted as an order, i.e. a numerical $\{<\}$-expansion of a structure.

Types Recall that for any logic $\mathrm{L} \in \{\mathrm{FO}, \mathrm{FO{+}MOD}, \mathrm{MSO}\}$ and $q \in \mathbb{N}$, we write $\mathfrak{A} \equiv_q^{\mathrm{L}} \mathfrak{B}$ for $q \in \mathbb{N}$ if the structures $\mathfrak{A}$ and $\mathfrak{B}$ satisfy the same L-sentences of quantifier rank at most q. The $\equiv_q^{\mathrm{L}}$-equivalence class of $\mathfrak{A}$ is its (L, q)-*type* and is denoted by $\mathrm{tp}_{\mathrm{L},q}(\mathfrak{A})$. We omit L in this notation whenever this is unambigously possible.

We often use the fact that two structures $\mathfrak{A}$ and $\mathfrak{B}$ have the same (FO, q)-type iff the Duplicator has a winning strategy in the q-round *Ehrenfeucht-Fraïssé game* on $\mathfrak{A}$ and $\mathfrak{B}$. We also use the extension of this game which captures (MSO, q)-types in the same way. For an explanation of both games, see e.g. [Lib04].

Relativisation of formulae If $\varphi(\bar{x})$ is a formula and $\psi(\bar{y}, z)$ is a formula with at least one free variable z, then $(\varphi \restriction \psi)(\bar{x}, \bar{y})$ is the *relativisation of* φ *to* ψ. We construct $\varphi \restriction \psi$ inductively by replacing subformulae of the form $\exists z\, \chi(\bar{x})$ and $\forall z\, \chi(\bar{x})$ by $\exists z\, (\psi(\bar{y}, z) \wedge \chi \restriction \psi(\bar{x}, \bar{y}))$ and $\forall z\, (\psi(\bar{y}, z) \to \chi \restriction \psi(\bar{x}, \bar{y}))$, respectively. Note that, if ψ is an existential formula, i.e. a formula which uses no $\forall$-quantifiers, then $\mathrm{qad}(\varphi|_\psi) = \mathrm{qad}(\varphi)$. To see that this is, in particular, true for the $\forall$-quantifier case of the construction, note that $(\psi(\bar{y}, z) \to \chi \restriction \psi(\bar{x}, \bar{y})) \equiv (\neg\psi(\bar{y}, z) \vee \chi \restriction \psi(\bar{x}, \bar{y}))$ where $\neg\psi(\bar{y}, z)$ is equivalent to a universal formula, i.e. a formula which uses no $\exists$-quantifiers, if $\psi(\bar{y}, z)$ is existential.

5.3.1 Encoding information about elements in extended signatures

In our proofs we will repeatedly remove single elements r from structures $\mathfrak{A}$ and encode information about the relations between r and the remaining elements into an expansion $\mathfrak{A}^{[r]}$ of the structure $\mathfrak{A} \setminus r$ (which is the substructure of $\mathfrak{A}$ induced on the elements different from r). We do this in such a way that the q-type of $\mathfrak{A}$ is determined by the q-type of $\mathfrak{A}^{[r]}$ together with, what we call, the atomic type of r in $\mathfrak{A}$.

The *atomic type* $\alpha(\mathfrak{A}, a)$ of an element a of a σ-structure $\mathfrak{A}$ is the set of all $R \in \sigma$ such that $(a, \dots, a) \in R^{\mathfrak{A}}$, where $(a, \dots, a)$ has length $\mathrm{ar}(R)$. If no confusion seems likely we omit $\mathfrak{A}$ and we write just $\alpha(a)$. Thus an atomic type is a subset of σ, and we identify $\alpha \subseteq \sigma$ with the $\mathrm{FO}[\sigma]$-sentence

$$\alpha(x) \;:=\; \bigwedge_{R \in \alpha} R(x, \dots, x) \;\wedge\; \bigwedge_{R \in \sigma \setminus \alpha} \neg R(x, \dots, x).$$

Since we will often need the atomic type of the $\leqslant$-minimal element of a structure, we denote by $\alpha_{\mathfrak{A}}$ the type $\alpha(r, \mathfrak{A})$ if $\mathfrak{A}$ is an ordered structure with minimal element r.

To encode the relations between the element which is removed and the remaining elements, we define a signature $\tilde{\sigma}$ which contains, for each $R \in \sigma$ and each nonempty $I \subseteq [1, \mathrm{ar}(R)]$, a relation symbol R_I of arity $|I|$. Given a structure $\mathfrak{A} = (A, (R^{\mathfrak{A}})_{R \in \sigma})$ and an element $r \in A$, we now obtain a $\tilde{\sigma}$-structure $\mathfrak{A}^{[r]} = (A \setminus \{r\}, (R_I^{\mathfrak{A}^{[r]}})_{R_I \in \tilde{\sigma}})$ by setting

$$R_I^{\mathfrak{A}^{[r]}} := \{(a_i)_{i \in I} : (a_1, \dots, a_{\mathrm{ar}(R)}) \in R^{\mathfrak{A}} \text{ and } a_i = r \text{ for } i \notin I\}.$$

Note that $R^{\mathfrak{A}} = R^{\mathfrak{A}^{[r]}}_{[1, \mathrm{ar}(R)]}$, so up to a renaming of relation symbols, $\mathfrak{A}^{[r]}$ is an expansion of $\mathfrak{A} \setminus r$.

The (L, q)-type of $\mathfrak{A}$ is determined by $\alpha(r)$ and the (L, q)-type of $\mathfrak{A}^{[r]}$:

Lemma 5.3.1. *Let* $\mathrm{L} \in \{\mathrm{FO}, \mathrm{MSO}\}$ *and* $q \in \mathbb{N}^+$. *Let* $\mathfrak{A}$ *and* $\mathfrak{B}$ *be structures,* $r \in A$ *and* $s \in B$. *If*

$$\alpha(\mathfrak{A}, r) = \alpha(\mathfrak{B}, s) \quad \textit{and} \quad \mathrm{tp}_{\mathrm{L},q}(\mathfrak{A}^{[r]}) = \mathrm{tp}_{\mathrm{L},q}(\mathfrak{B}^{[s]}),$$

then also

$$\mathrm{tp}_{\mathrm{L},q}(\mathfrak{A}) = \mathrm{tp}_{\mathrm{L},q}(\mathfrak{B}).$$

Proof. The same argument works for $\mathrm{L} = \mathrm{FO}$ and $\mathrm{L} = \mathrm{MSO}$. Duplicator has a winning strategy $\mathcal{S}$ in the q-round Ehrenfeucht-Fraïssé game for L on $\mathfrak{A}^{[r]}$ and $\mathfrak{B}^{[s]}$. Note that the strategy $\mathcal{S}$ is, in particular, a winning strategy on $\mathfrak{A} \setminus r$ and $\mathfrak{B} \setminus s$, because $\mathfrak{A}^{[r]}$ and $\mathfrak{B}^{[s]}$ are expansions of these structures. Duplicator can win the q-round EF-game on $\mathfrak{A}$ and $\mathfrak{B}$ if she plays according to $\mathcal{S}$ on $\mathfrak{A} \setminus r$ and $\mathfrak{B} \setminus s$, and if she responds to r with s and vice versa.

We have to argue that this strategy preserves relations between the played elements. For relations not involving the removed elements r and s, this is true because S is a winning strategy for the q-round game on $\mathfrak{A} \setminus r$ and $\mathfrak{B} \setminus s$. Relations involving only the minimal elements are preserved because $\alpha(\mathfrak{A}, r) = \alpha(\mathfrak{B}, s)$. Relations involving the minimal elements and other elements are preserved, because they are encoded in the relations R_I of the extended signature $\tilde{\sigma}$, and these are preserved by $\mathcal{S}$. □

The following lemma is easy to prove following these definitions:

Lemma 5.3.2. *Let* $\mathrm{L} \in \{\mathrm{FO}, \mathrm{FO{+}MOD}\}$. *For every* $\mathrm{L}[\tilde{\sigma}]$*-sentence* φ *there is an* $\mathrm{L}[\sigma]$*-formula* $\mathcal{I}(\varphi)(z)$ *of the same quantifier rank and quantifier alternation depth such that*

$$\mathfrak{A} \models \mathcal{I}(\varphi)(r) \quad \textit{iff} \quad \mathfrak{A}^{[r]} \models \varphi,$$

for all σ*-structures* $\mathfrak{A}$ *and* $r \in A$.

Proof. The proof uses a standard interpretation argument. It suffices to provide quantifier-free formulae with a parameter z which define the universe and the relations of $\mathfrak{A}^{[r]}$ in $\mathfrak{A}$, provided that z is interpreted by the element r. The universe is defined by the formula $x \neq z$. Let $R_I \in \tilde{\sigma}$. If, for each $i \leqslant \mathrm{ar}(R)$, we let

$$y_i := \begin{cases} x_j & \text{if } i = i_j \in I \\ z & \text{if } i \notin I \end{cases}$$

then $R(y_1, \ldots, y_{\mathrm{ar}(R)})$ is a formula with free variables $z, x_1, \ldots, x_{|I|}$ which defines $R_I^{\mathfrak{A}^{[r]}}$ in $(\mathfrak{A}, r)$. □

5.3.2 Tree-depth

The following inductive definition is one of several equivalent ways to define the *tree-depth* $\mathrm{td}(G)$ of a graph (see [NdM12] for a reference on tree-depth):

$$\mathrm{td}(G) := \begin{cases} 1 & \text{if } |V(G)| = 1 \\ 1 + \min_{r \in V(G)} \mathrm{td}(G \setminus r) & \text{if } G \text{ is connected and } |V(G)| > 1 \\ \max_{i \in [1,n]} \mathrm{td}(K_i) & \text{if } G \text{ has components } K_1, \ldots, K_n. \end{cases}$$

As usual, the tree-depth $\mathrm{td}(\mathfrak{A})$ of a relational structure $\mathfrak{A}$ is defined by

$$\mathrm{td}(\mathfrak{A}) := \mathrm{td}(\mathfrak{G}(\mathfrak{A})).$$

For each $d \in \mathbb{N}^+$, we let

$$\begin{aligned} \mathrm{FIN}_{\sigma,d} &:= \{\mathfrak{A} \in \mathrm{FIN}_\sigma : \mathrm{td}(\mathfrak{A}) \leqslant d\}, \\ \mathrm{FIN}_d &:= \bigcup_\sigma \mathrm{FIN}_{\sigma,d}, \end{aligned}$$

and

$$\begin{aligned} \mathrm{FIN}_\sigma^{\mathrm{conn}} &:= \{\mathfrak{A} \in \mathrm{FIN}_\sigma : \mathfrak{A} \text{ is connected}\}, \\ \mathrm{FIN}_{\sigma,d}^{\mathrm{conn}} &:= \{\mathfrak{A} \in \mathrm{FIN}_\sigma^{\mathrm{conn}} : \mathrm{td}(\mathfrak{A}) \leqslant d\}. \end{aligned}$$

As an immediate consequence of the above definition of tree-depth, for each $\mathfrak{A} \in \mathrm{FIN}_{\sigma,d}^{\mathrm{conn}}$, either $\mathrm{td}(\mathfrak{A}) = 1$ and $\mathfrak{A}$ contains only one element or $\mathrm{td}(\mathfrak{A}) > 1$ and $\mathfrak{A}$ contains an element whose removal reduces the tree-depth of $\mathfrak{A}$. We call these elements *tree-depth roots* and denote the set of all such elements by $\mathrm{roots}(\mathfrak{A})$. That is, $r \in \mathrm{roots}(\mathfrak{A})$ iff either $A = \{r\}$ or $|A| > 1$ and $\mathrm{td}(\mathfrak{A} \setminus r) \leqslant \mathrm{td}(\mathfrak{A}) - 1$. By the following result of Bouland, Dawar, and Kopczyński [BDK12], for each graph G the size of $\mathrm{roots}(G)$ is bounded by a function of its tree-depth which is independent of the size of G (and hence the same is true for structures).

Lemma 5.3.3 ([BDK12, Lem. 7]). *There is a function $f : \mathbb{N}^+ \to \mathbb{N}^+$ such that $|\mathrm{roots}(G)| \leqslant f(\mathrm{td}(G))$ for each connected graph G.*

Note that the definition of $\mathrm{roots}(G)$ in [BDK12] is slightly different from ours, but the two definitions are easily seen to be equivalent.

A graph of tree-depth at most d can not contain a path of length 2^d (cf. [NdM12, 6.2]). Therefore $\mathrm{dist}^{\mathfrak{A}}(a,b) < 2^d$ for all elements a and b in the same connected component of a structure $\mathfrak{A}$ of tree-depth at most d. It is plain that there is an existential FO[σ]-formula $\mathrm{dist}_{\leqslant \ell}(x,y)$ such that $\mathfrak{A} \models \mathrm{dist}_{\leqslant \ell}(a,b)$ iff $\mathrm{dist}^{\mathfrak{A}}(a,b) \leqslant \ell$, for each σ-structure $\mathfrak{A}$ and all $a, b \in A$. Hence, the formula

$$\mathrm{reach}_d(x,y) := \mathrm{dist}_{<2^d}(x,y)$$

defines the reachability relation in structure of tree-depth at most d. That is,

$$\mathfrak{A} \models \mathrm{reach}_d[a,b] \quad \text{iff} \quad a \text{ and } b \text{ belong to the same component of } \mathfrak{A}.$$

This existential formula allows us to relativise a formula $\varphi(x)$ to the connected component of x:

$$\mathfrak{A} \models \varphi \restriction \mathrm{reach}_d(x,z)[a] \quad \text{iff} \quad \mathfrak{K} \models \varphi[a],$$

where $\mathfrak{K}$ is (the substructure of $\mathfrak{A}$ induced on) the connected component of a in $\mathfrak{A}$. Since reach_d is existential, we have $\mathrm{qad}(\varphi \restriction \mathrm{reach}_d(x,z)) = \mathrm{qad}(\varphi)$.

Using these observations and the inductive definition of tree-depth, it is easy to write down an FO[σ]-sentence that defines $\mathrm{FIN}_{\sigma,d}$ on the class of all finite σ-structures. While this naïve approach leads to a formula whose quantifier alternation depth grows linearly with d, it is also possible to construct a *universal* sentence $\mathrm{td}_{\leqslant d}$ defining $\mathrm{FIN}_{\sigma,d}$ as a subclass of FIN_{σ}, cf. [NdM12, Section 6.10] for details. Using this sentence, we construct a sentence that defines the set $\mathrm{roots}(\mathfrak{A})$ for each $\mathfrak{A} \in \mathrm{FIN}^{\mathrm{conn}}_{\sigma,d}$. To this end, we let $\mathrm{roots}_1(x) := \forall x\, x = x$ and $\mathrm{roots}_d(x) := \bigvee_{c \leqslant d-1} \big(\mathrm{td}_{>c} \wedge \mathrm{td}_{\leqslant c}\!\upharpoonright\!(x \neq z)(x)\big)$ if $d > 1$.

5.4 Order-invariant FO

It is well-known that order-invariance is undecidable on the class FIN (cf. [Gur88]), i.e. there is no algorithm which decides for a given FO[$\sigma, <$]-sentence if it is order-invariant on FIN. This leads to the question if the expressive power of order-invariant sentences on a class $\mathcal{C}$ can be captured by a logic with an effective syntax. An answer to this question in the case of the class FIN seems out of reach. We consider the question in the case of *bounded tree-depth structures*, i.e. $\mathcal{C} = \mathrm{FIN}_d$ for some $d \in \mathbb{N}^+$. More concretely, our aim is a proof of the following theorem:

Theorem 5.4.1. *For each $d \in \mathbb{N}^+$, each signature σ, and each $<$-inv-FO[σ]-sentence φ, there is an FO[σ]-sentence ψ which is equivalent to φ on $\mathrm{FIN}_{\sigma,d}$ and which has size $|\psi| \in \mathrm{EXP}_d(\mathrm{qr}(\varphi))$ and quantifier-alternation depth $\mathrm{qad}(\psi) \leqslant 3d$.*

The proof of Theorem 5.4.1 will be presented in Section 5.4.2 below. Before that, we want to motivate Theorem 5.4.1 by showing that the undecidability of order-invariance holds even for structures of tree-depth 2.

5.4.1 Undecidability on graphs of tree-depth 2

As mentioned by Schweikardt [Sch13], order-invariance for formulae over a signature σ which contains only unary relation symbols besides the symbol for the order is decidable. An ordered σ-structure in which the unary relations partition the universe can be regarded as a word. An FO[$\sigma, <$]-sentence φ then defines a language L_φ. The sentence φ is order-invariant iff the syntactic monoid of L_φ is commutative, which is decidable. Using simple interpretation arguments, it can be seen that this decidability result extends to general σ-structures (where the unary relations are not necessarily disjoint) and to structures of tree-depth 1 over arbitrary signatures.

Hence, order-invariance is decidable on FIN_d if $d = 1$. The next theorem shows that it becomes undecidable for $d \geqslant 2$.

Theorem 5.4.2. *Order-invariance is undecidable on* FIN_2.

The proof of Theorem 5.4.2 uses a reduction from the undecidable halting problem for *counter machines* (cf. [Min67]) with two counters which store natural numbers. A counter machine executes a *program*, i.e. a finite sequence of the following *instructions*:

INC(i) Increment counter i, proceed with next instruction.

DEC(i, j_0, j_1)

If counter i is not zero: decrement counter i, proceed with j_1-th instruction.

Otherwise: proceed with instruction j_0.

HALT Stop the execution.

The configuration of the machine at any execution step is fully described by a triple (n_1, n_2, j), where $n_1, n_2 \geqslant 0$ are natural numbers stored in the counters and $j \geqslant 1$ is the number of the next instruction to be executed. Without loss of generality, we assume that the last instruction of a program is always the HALT instruction and that this instruction occurs nowhere else in the program. Hence, we say that a program *halts* (on empty input) if it ever reaches its last instruction when run from the initial configuration $(0, 0, 1)$.

Proof of Theorem 5.4.2. We say that a sentence $\varphi \in \mathrm{FO}[\sigma, <]$ is *d-satisfiable* if it has a model $(\mathfrak{A}, <^{\mathfrak{A}})$ where $\mathfrak{A} \in \mathrm{FIN}_{\sigma,d}$. The folklore proof which shows that order-invariance on FIN is undecidable uses a many-one reduction from the undecidable finite satisfiability problem to order-invariance. The same kind of argument proves that *d-satisfiability* (i.e. the problem which asks if an input sentence $\varphi \in \mathrm{FO}[\sigma, <]$ over an arbitrary relational signature σ is d-satisfiable) many-one reduces to order-invariance on $\mathrm{FIN}_{\tilde{\sigma},d}$, where $\tilde{\sigma} := \sigma \cup \{P\}$ for a unary relation symbol $P \notin \sigma$. This follows from the fact that $\varphi \in \mathrm{FO}[\sigma, <]$ is d-satisfiable iff the $\mathrm{FO}[\tilde{\sigma}^{\leqslant}]$-sentence $\varphi \wedge \exists x \forall y\ (x<y\ \wedge P(x))$ is *not* order-invariant on $\mathrm{FIN}_{\tilde{\sigma},d}$.

Hence, to complete the proof of our theorem, it suffices to show that the 2-satisfiability problem is undecidable. To this end we reduce the halting problem for counter machines to 2-satisfiability. Let $P = I_1 \cdots I_\ell$ be a program. We construct an $\mathrm{FO}[\sigma, <]$-sentence φ which is $\mathrm{FIN}_{\sigma,2}$-satisfiable iff P halts, for some signature σ which depends on ℓ. First we fix an encoding of configurations of P by words over a finite alphabet Σ. It would be natural to do this by encoding the counter values in unary using different symbols; say, $(2, 3, 1)$ would become $\mathbf{11}\,\mathbf{222}\,1$. We change this representation slightly: a configuration (n_1, n_2, j) of P is encoded by a word

$$\mathrm{enc}(n_1, n_2, j) \ := \ (\mathbf{1}_L \mathbf{1}_R)^{n_1}\, (\mathbf{2}_L \mathbf{2}_R)^{n_2}\, j$$

over the alphabet $\Sigma := \{\mathbf{1}_L, \mathbf{2}_L, \mathbf{1}_R, \mathbf{2}_R, 1, \dots, \ell\}$.

Let $\sigma := \{E\} \cup \tau$ where E is a binary relation symbol and $\tau := \{P_a : a \in \Sigma\}$, where the P_a are unary relation symbols. The σ-structures that we consider are *Σ-coloured graphs* (i.e. E is used for the edge relation and the P_a for a partition of the vertex set). If a vertex of such a graph belongs to a relation P_a, we say that it is *a-coloured*. The class of Σ-coloured graphs is obviously FO-definable on FIN_σ.

As usual, we identify each non-empty word over the alphabet Σ with an ordered τ-structure which, in turn, we regard as an ordered Σ-coloured graph with no edges. We refer to vertices which are coloured by $1, \dots, \ell$ as *instruction vertices*. If our program P halts after at most h computation steps then, with respect to our encoding, there exists a unique word w_P which encodes the *run* of P, i.e. the finite sequence of configurations at time steps $1, \ldots, h$. We want to define a class of ordered Σ-coloured graphs of maximum degree 1 obtained from the edge-less graph w_P by adding edges between its vertices. These graphs will be called *matching extensions* of w_P, since their edge relations will be unions of matchings (i.e. edge relations of graphs where each vertex is incident to exactly one edge). Consider any word $w = \mathrm{enc}(C_1) \cdots \mathrm{enc}(C_k)$ which encodes a sequence of representations. We phrase the description of the execution of the counter machine program P given in the definition of counter machines above somewhat more formally as conditions under which the sequence $C_1, \ldots, C_k$ is a run of P (i.e. $w = w_P$). At the same time, we rephrase them as statements about the ordered Σ-coloured graph w in a way that will be suitable for the definition of our sentence φ.

1. $C_1 = (0, 0, 1)$ and C_k is a halting configuration, i.e. $C_k = (n_1, n_2, \ell)$ for some $n_1, n_2 \geqslant 0$.

   With our encoding, this is equivalent to the first vertex of w being 1-coloured and the last vertex being ℓ-coloured. (Recall that the machine starts with both counters being 0.)

2. For each $i \in [1, k]$ and $C_i = (n_1, n_2, j)$ one of the following statements is true:

   (a) $I_j = \textsc{inc}(1)$ and $C_{i+1} = (n_1 + 1, n_2, j + 1)$.

   This holds iff we can add edges to w so that all $\mathbf{1}_L$-coloured vertices in $\mathrm{enc}(C_i)$ are matched with all but one of the $\mathbf{1}_R$-coloured vertices in $\mathrm{enc}(C_{i+1})$, and the $\mathbf{2}_L$-coloured vertices in $\mathrm{enc}(C_i)$ are matched with the $\mathbf{2}_R$-coloured vertices in $\mathrm{enc}(C_{i+1})$, and if the unique instruction vertex of $\mathrm{enc}(C_i)$ is j-coloured, for some $j \in [1, \ell]$, then the unique instruction vertex of $\mathrm{enc}(C_{i+1})$ is $(j + 1)$-coloured.

   (b) $I_j = \textsc{dec}(1, j_0, j_1)$ and either $n_1 = 0$ and $C_{i+1} = (n_1, n_2, j_0)$, or $n_1 \geqslant 1$ and $C_{i+1} = (n_1 - 1, n_2, j_1)$.

   Equivalently, either one of the following statements is true:

   - There exists no $\mathbf{1}_L$-coloured vertex in $\mathrm{enc}(C_i)$ and no $\mathbf{1}_L$-coloured vertex in $\mathrm{enc}(C_{i+1})$. Furthermore, the $\mathbf{2}_L$-coloured vertices in $\mathrm{enc}(C_i)$ can be matched with the $\mathbf{2}_R$-coloured vertices in $\mathrm{enc}(C_{i+1})$. The unique instruction vertex in $\mathrm{enc}(C_{i+1})$ is j_0-coloured.
   - There is at least one $\mathbf{1}_L$-coloured vertex in $\mathrm{enc}(C_i)$. Furthermore, the $\mathbf{1}_R$-coloured vertices in $\mathrm{enc}(C_{i+1})$ can be matched with all but one of the $\mathbf{1}_L$-coloured vertices in $\mathrm{enc}(C_i)$, and the $\mathbf{2}_L$-coloured vertices in $\mathrm{enc}(C_i)$ can be matched with the $\mathbf{2}_R$-coloured vertices in $\mathrm{enc}(C_{i+1})$. The unique instruction vertex in $\mathrm{enc}(C_{i+1})$ is j_1-coloured.

   (c),(d) Analogous statements to (a), (b) for the case where I_j operates on counter 2.

Now, a *matching extension* of w_P is an ordered graph obtained from w_P by adding, for each pair of subsequent configurations, exactly the edges of a matching witnessing that w_P satisfies the conditions (a), (b), (c), and (d). Observe that each vertex of a matching extension is contained in at most one matching. Hence, any matching extension has maximum degree 1. Using our description above, it is easy to write down a first-order sentence φ defining the class of all matching extensions of w_P. This class is non-empty iff P halts. Hence, φ is 2-satisfiable iff P halts. $\square$

The alphabet in the proof and hence the signature σ depends on the length of the given program P. The proof can be modified easily to make the alphabet Σ, and therefore the signature σ, independent of P without increasing the tree-depth of the structures involved. To this end, we would introduce a single new unary relation symbol N to the signature. The labelling of a vertex v by a number $i \in [1, \ell]$ can be encoded by adding i new pendant vertices as neighbours of v, marked by N. This does not increase the tree-depth since the resulting graph (without the order) is still a disjoint union of stars. The vertices belonging to N can, e.g., be placed after all remaining vertices in the linear order. This shows that order-invariance is already undecidable on the class $\mathrm{FIN}_{\sigma,2}$, for a fixed signature σ.

5.4.2 From order-invariant FO to FO

We prove Theorem 5.4.1. The key insight here is that for every quantifier rank q and every structure $\mathfrak{A} \in \mathrm{FIN}_{\sigma,d}$ there exists a class of canonical linear orders $\preceq_q$ for which the FO_q-type of $(\mathfrak{A}, \preceq_q)$ is already FO-definable in $\mathfrak{A}$. In particular, $\mathrm{tp}_q(\mathfrak{A}, \preceq_q)$ only depends on $\mathfrak{A}$, even though there may be more than one such order on $\mathfrak{A}$.

We call these canonical orders *q-orders*. After defining them formally, we prove the following two facts about them:

1. Expansions by q-orders are indistinguishable by FO-sentences of quantifier rank q, i.e. $(\mathfrak{A}, \preceq_1) \equiv_q (\mathfrak{A}, \preceq_2)$ for all finite structures $\mathfrak{A}$, provided both $\preceq_1$ and $\preceq_2$ are q-orders (cf. Lemma 5.4.4).

2. If the tree-depth of structures is bounded, then the q-type $\mathrm{tp}_q(\mathfrak{A}, \preceq_q)$ of an expansion of $\mathfrak{A}$ by a q-order is definable in FO (Lemmas 5.4.8 and 5.4.11). The proof of Theorem 5.4.1 easily follows from this.

The definition of q-orders With an eye towards Section 5.5, the notion of q-orders will be defined more generally for logics $\mathrm{L} \in \{\mathrm{FO}, \mathrm{MSO}\}$. We fix arbitrary orders $\preceq_{\mathrm{L},q}$ on the set of (L, q)-types over the signature $(\sigma, <)$, and $\preceq_{\text{atomic}}$ on the set of atomic σ-types. For simplicity, we write $a \preceq_{\text{atomic}} b$ for $\alpha(a) \preceq_{\text{atomic}} \alpha(b)$.

To obtain a q-order $\preceq$ on a connected structure $\mathfrak{A} \in \mathrm{FIN}_{\sigma,d}$, we pick a root r of $\mathfrak{A}$ which has $\preceq_{\text{atomic}}$-minimal atomic type among all roots and for which the type of q-ordered expansions of $\mathfrak{A}^{[r]}$ is $\preceq_{\mathrm{L},q}$-minimal among all $\preceq_{\text{atomic}}$-minimal roots. We place this r in front of the order $\preceq$ and order the remaining elements according to a (recursively obtained) q-order on $\mathfrak{A}^{[r]}$. On structures with more than one component, we q-order the components individually and take the sum of their orders, following the $\preceq_{\mathrm{L},q}$-order of the components:

Definition 5.4.3 ((L, q)-order)**.** An (L, q)-*order* on a σ-structure $\mathfrak{A}$ is an order $\preceq$ which satisfies the following conditions:

(1) If $\mathfrak{A}$ is connected we denote by $r \in A$ its $\preceq$-minimal element. Then either $|A| = 1$, or $|A| > 1$ and the following holds:

(a) r is a $\preceq_{\text{atomic}}$-minimal root of $\mathfrak{A}$, i.e. $r \in \mathrm{roots}(\mathfrak{A})$ and $r \preceq_{\text{atomic}} r'$ for all $r' \in \mathrm{roots}(\mathfrak{A})$.

(b) The (L, q)-type of q-ordered expansions of $\mathfrak{A}^{[r]}$ is minimal:

$$\mathrm{tp}_q(\mathfrak{A}^{[r]}, \preceq) \preceq_{\mathrm{L},q} \mathrm{tp}_q(\mathfrak{A}^{[r']}, \preceq')$$

for every $r' \in \mathrm{roots}(\mathfrak{A})$ with $\alpha(r') = \alpha(r)$ and every q-order $\preceq'$ on $\mathfrak{A}^{[r']}$.

(c) $\preceq \restriction A \setminus r$ is an (L, q)-order on $\mathfrak{A}^{[r]}$.

(2) If $\mathfrak{A}$ is not connected, we denote its components by $\mathfrak{A}_1, \ldots, \mathfrak{A}_\ell$ and set $\preceq_i := \preceq \restriction A_i$. Then $\preceq$ is a q-order if

(a) each $\preceq_i$ is a q-order of $\mathfrak{A}_i$, and

(b) after suitably permuting the components,

$$\preceq = \preceq_1 + \cdots + \preceq_\ell \quad \text{and} \quad \mathrm{tp}_q(\mathfrak{A}_i, \preceq_i) \preceq_{\mathrm{L},q} \mathrm{tp}_q(\mathfrak{A}_j, \preceq_j) \text{ for } i \leqslant j.$$

The $\preceq$-minimal element of a q-order $\preceq$ will be denoted by $r_\preceq$.

It is plain from the definition above that each structure can be q-ordered. Next we want to show that all q-ordered expansions $(\mathfrak{A}, \preceq)$ of a given structure $\mathfrak{A}$ have the same q-type, and that the q-type of $(\mathfrak{A}^{[r_\preceq]}, \preceq)$ is also the same for all q-orders $\preceq$ of $\mathfrak{A}$.

Lemma 5.4.4. *Let* $\mathrm{L} \in \{\mathrm{FO}, \mathrm{MSO}\}$, $q \in \mathbb{N}^+$. *For all* (L, q)*-orders* $\preceq, \preceq'$ *of a structure* $\mathfrak{A}$, *we have*

$$(\mathfrak{A}, \preceq) \equiv_q^{\mathrm{L}} (\mathfrak{A}, \preceq').$$

If $\mathfrak{A}$ *is connected and* $\mathrm{td}(A) > 1$, *then also* $(\mathfrak{A}^{[r_\preceq]}, \preceq) \equiv_q^{\mathrm{L}} (\mathfrak{A}^{[r_{\preceq'}]}, \preceq')$.

For the proof, we need the following folklore composition lemma for ordered sums which has a simple EF-game-based proof (a proof of a somewhat stronger statement can also be found in [Mak04]).

Lemma 5.4.5 (Composition Lemma). *Let* $\mathrm{L} \in \{\mathrm{FO}, \mathrm{MSO}\}$, $q \in \mathbb{N}$ *and let* σ *be a relational signature. Let* $(\mathfrak{A}_1, \preceq^{\mathfrak{A}_1})$, $(\mathfrak{A}_2, \preceq^{\mathfrak{A}_2})$, $(\mathfrak{B}_1, \preceq^{\mathfrak{B}_1})$, $(\mathfrak{B}_2, \preceq^{\mathfrak{B}_2})$ *be ordered* σ*-structures. If*

$$(\mathfrak{A}_1, \preceq^{\mathfrak{A}_1}) \equiv_q^{\mathrm{L}} (\mathfrak{A}_2, \preceq^{\mathfrak{A}_2}) \quad \textit{and} \quad (\mathfrak{B}_1, \preceq^{\mathfrak{B}_1}) \equiv_q^{\mathrm{L}} (\mathfrak{B}_2, \preceq^{\mathfrak{B}_2}),$$

then

$$(\mathfrak{A}_1 \sqcup \mathfrak{B}_1, \preceq^{\mathfrak{A}_1} + \preceq^{\mathfrak{B}_1}) \equiv_q^{\mathrm{L}} (\mathfrak{A}_2 \sqcup \mathfrak{B}_2, \preceq^{\mathfrak{A}_2} + \preceq^{\mathfrak{B}_2}).$$

Proof of Lemma 5.4.4. The proof proceeds on the size of A. If $|A| = 1$ then $\preceq = \preceq'$ and there is nothing to prove.

Let $|A| > 1$ and suppose first that $\mathfrak{A}$ is connected. By Definition 5.4.3, $\alpha(r_\preceq) = \alpha(r_{\preceq'})$ and

$$\mathrm{tp}_q(\mathfrak{A}^{[r_\preceq]}, \preceq) \preceq_{\mathrm{L},q} \mathrm{tp}_q(\mathfrak{A}^{[r_{\preceq'}]}, \preceq').$$

By symmetry also

$$\mathrm{tp}_q(\mathfrak{A}^{[r_{\preceq'}]}, \preceq') \preceq_{\mathrm{L},q} \mathrm{tp}_q(\mathfrak{A}^{[r_\preceq]}, \preceq),$$

so $\mathrm{tp}_q(\mathfrak{A}^{[r_\preceq]}, \preceq) = \mathrm{tp}_q(\mathfrak{A}^{[r_{\preceq'}]}, \preceq')$ and, by Lemma 5.3.1, $(\mathfrak{A}, \preceq) \equiv_q (\mathfrak{A}, \preceq')$.

Now consider the case where $\mathfrak{A}$ is not connected, and let $\mathfrak{K}_1, \dots, \mathfrak{K}_\ell$ be the components of $\mathfrak{A}$. By the definition of q-orders each $\mathfrak{K}_i$ is q-ordered, so

$$(\mathfrak{K}_i, \preceq \restriction K_i) \equiv_q^{\mathrm{L}} (\mathfrak{K}_i, \preceq' \restriction K_i)$$

for $i = 1, \dots, \ell$ by what we have just said. Considering the way that an (L, q)-order orders the components of a structure according to their (L, q)-types (part 2 of Definition 5.4.3), we obtain that $(\mathfrak{A}, \preceq) \equiv_q^{\mathrm{L}} (\mathfrak{A}, \preceq')$ by repeatedly applying the Composition Lemma. □

By Lemma 5.4.4 it makes sense to speak of the *q-order type* of an unordered structure $\mathfrak{A}$ which we define as

$$\mathrm{tp}_q^<(\mathfrak{A}) := \mathrm{tp}_q(\mathfrak{A}, \preceq_q).$$

If $\mathfrak{A}$ is connected and $\mathrm{td}(\mathfrak{A}) > 1$, we furthermore define its *q-order root type* as

$$\mathrm{rtp}_q^<(\mathfrak{A}) := \mathrm{tp}_q(\mathfrak{A}^{[r_{\preceq_q}]}, \preceq_q).$$

In both cases $\preceq_q$ is some q-order on $\mathfrak{A}$ and well-definedness is guaranteed by the Lemma. Note that both these types are $(\sigma, <)$-types. Similarly, the atomic type $\alpha_\mathfrak{A} := \alpha(r_\leqslant)$ of the minimal element in a q-ordered expansion of $\mathfrak{A}$ is well-defined.

We set

$$\begin{aligned} \mathcal{T}_{\mathrm{L},\sigma,q,d} &:= \{\mathrm{tp}_q^<(\mathfrak{A}) \mid \mathfrak{A} \in \mathbf{FIN}_{\sigma,d}\}, \\ \mathcal{T}^{\mathrm{conn}}_{\mathrm{L},\sigma,q,d} &:= \{\mathrm{tp}_q^<(\mathfrak{A}) \mid \mathfrak{A} \in \mathbf{FIN}^{\mathrm{conn}}_{\sigma,d}\}, \text{ and} \\ \mathcal{T}_{\mathrm{L},\sigma,q} &:= \bigcup_{d \in \mathbb{N}^+} \mathcal{T}_{\mathrm{L},q,\sigma,d}. \end{aligned}$$

We say that a sentence $\varphi_\tau \in \mathrm{L}[\sigma]$ *defines τ on* $\mathrm{FIN}_{\sigma,d}$ (and that τ is L-definable) if for each $\mathfrak{A} \in \mathbf{FIN}_{\sigma,d}$, we have

$$\mathfrak{A} \models \varphi_\tau \quad \text{iff} \quad \mathrm{tp}_q^<(\mathfrak{A}) = \tau.$$

Note that the sentence φ_τ must not contain the relation $<$.

By Lemma 5.3.1 the atomic type of $r_\preceq$ and the q-type of $\mathfrak{A}^{[r_\preceq]}$ determine the q-type of $\mathfrak{A}$, and $\mathrm{td}(\mathfrak{A}^{[r_\preceq]}) = \mathrm{td}(\mathfrak{A}) - 1$, for connected structures $\mathfrak{A}$ and q-orders $\preceq$. Since the number of atomic $\tilde\sigma$-types is $2^{|\tilde\sigma|}$, we obtain the following bound on the size of $\mathcal{T}^{\mathrm{conn}}_{\sigma,q,d}$:

Corollary 5.4.6. *Let $q, d \in \mathbb{N}^+$. Then $|\mathcal{T}^{conn}_{\sigma,q,d}| \leqslant 2^{|\tilde\sigma|} \cdot |\mathcal{T}_{\tilde\sigma,q,d-1}|$.*

Connected structures

The proof of our main theorem is broken down into two steps. In the first step, we show how to lift the definability of q-types of q-ordered structures from structures of tree-depth $d-1$ to connected structures of tree-depth d.

Again we invoke Lemma 5.3.1 and Lemma 5.4.4 to show that q-order types can be broken down into atomic types of roots and q-order root types:

Corollary 5.4.7. *Let $d > 1$ and let $\tau \in \mathcal{T}^{conn}_{\sigma,q,d}$. Let*

$$R_\tau \;:=\; \{(\alpha_{\mathfrak{A}}, \mathrm{rtp}^{<}_{q}(\mathfrak{A})) \mid \mathfrak{A} \in \mathrm{FIN}^{conn}_{\sigma,d}, \mathrm{td}(\mathfrak{A}) > 1,\ \textit{and}\ \mathrm{tp}^{<}_{q}(\mathfrak{A}) = \tau\}.$$

Then for each $\mathfrak{B} \in \mathrm{FIN}^{conn}_{\sigma,d}$, we have $\mathrm{tp}^{<}_{q}(\mathfrak{B}) = \tau$ iff $(\alpha_{\mathfrak{B}}, \mathrm{rtp}^{<}_{q}(\mathfrak{B})) \in R_\tau$.

Proof. The "only-if"-part of the claim is obvious. Regarding the "if"-part, if

$$(\alpha_{\mathfrak{B}}, \mathrm{rtp}^{<}_{q}(\mathfrak{B})) = (\alpha_{\mathfrak{A}}, \mathrm{rtp}^{<}_{q}(\mathfrak{A}))$$

for some $\mathfrak{A}$ with $\mathrm{tp}^{<}_{q}(\mathfrak{A}) = \tau$, then Lemma 5.4.4 and the definitions of $\mathrm{tp}^{<}_{q}, \mathrm{rtp}^{<}_{q}$ imply that $\mathrm{tp}^{<}_{q}(\mathfrak{B}) = \tau$. □

For the next lemma, recall that $\tilde{\sigma}$ is the signature of the structure $\mathfrak{A}^{[r]}$ if $\mathfrak{A}$ is a σ-structure (cf. page 112).

Lemma 5.4.8. *Let $q, d \in \mathbb{N}^+$ with $d > 1$. Let $(\mathrm{L}_1, \mathrm{L}_2)$ be one of $(\mathrm{FO}, \mathrm{FO})$ or $(\mathrm{MSO}, \mathrm{FO{+}MOD})$. If each (L_1, q)-type $\theta \in \mathcal{T}_{\tilde{\sigma},q,d-1}$ is $\mathrm{L}_2[\tilde{\sigma}]$-definable on $\mathrm{FIN}_{\tilde{\sigma},d-1}$ by a sentence $\psi_{\theta,d-1}$, then each (L_1, q)-type $\tau \in \mathcal{T}^{conn}_{\sigma,q,d}$ is $\mathrm{L}_2[\sigma]$-definable on $\mathrm{FIN}^{conn}_{\sigma,d}$ by a sentence $\varphi^{conn}_{\tau,d}$. Moreover, defining*

$$\Psi := \{\psi_{\theta,d-1} : \theta \in \mathcal{T}_{\tilde{\sigma},q,d-1}\} \quad \textit{and} \quad \Phi := \{\varphi^{conn}_{\tau,d} : \tau \in \mathcal{T}^{conn}_{\sigma,q,d}\},$$

we have $|\Phi| \leqslant c \cdot |\Psi| \cdot |\mathcal{T}_{\tilde{\sigma},q,d-1}|^2$ and $\mathrm{qad}(\Psi) \leqslant \mathrm{qad}(\Phi) + 1$, for a constant c depending only on σ, d.

Proof. In the following, all q-types are $(\mathrm{L}_1, (\sigma, <), q)$-types. Let $\tau \in \mathcal{T}^{\mathrm{conn}}_{\sigma,q,d}$ and let R_τ be as in Corollary 5.4.7. We show that, under the assumptions of our lemma, the class

$$\{\mathfrak{A} \in \mathrm{FIN}^{\mathrm{conn}}_{\sigma,d} : (\alpha_{\mathfrak{A}}, \mathrm{rtp}^{<}_{q}(\mathfrak{A})) \in R_\tau\}$$

is $\mathrm{L}_2[\sigma]$-definable by a sentence φ_τ on $\mathrm{FIN}^{\mathrm{conn}}_{\sigma,d}$. Taking care of connected structures of tree-depth 1 (i.e. singleton structures), we set $\varphi^{\mathrm{conn}}_{\tau,d} \;:=\; (\mathrm{td}_{\leqslant 1} \wedge \hat{\varphi}_\tau) \vee (\mathrm{td}_{>1} \wedge \varphi_\tau)$, where $\hat{\varphi}_\tau$ defines τ on singleton structures.

For each atomic σ-type $\alpha \subseteq \sigma$, the following FO-sentence ξ_α expresses in a structure $\mathfrak{A} \in \mathrm{FIN}^{\mathrm{conn}}_{\sigma,d}$ that $\alpha_{\mathfrak{A}} = \alpha$:

$$\xi_\alpha \;:=\; \big(\exists x \left(\mathrm{roots}_d(x) \wedge \alpha(x)\right)\big) \wedge \left(\forall x \left(\mathrm{roots}_d(x) \to \bigvee_{\alpha \preceq_{\mathrm{atomic}} \alpha'} \alpha'(x)\right)\right).$$

For each type $\theta \in \mathcal{T}_{\tilde{\sigma},q,d-1}$ the following sentence is true in a σ-structure $\mathfrak{A}$ iff there is a root r of atomic type α for which $\mathfrak{A}^{[r]}$ has type θ, and θ is $\preceq_{\mathrm{L}_1,q}$-minimal among the types of $\mathfrak{A}^{[s]}$ for roots s of atomic type α:

$$\begin{aligned}\chi_{\alpha,\theta} \;:=\; &\forall x \Big((\mathrm{roots}_d(x) \wedge \alpha(x)) \to \bigvee_{\theta \preceq_{\mathrm{L}_1,q} \theta'} \mathcal{I}(\psi_{\theta',d-1})(x) \Big) \\ &\wedge \exists x \, \big(\mathrm{roots}_d(x) \wedge \alpha(x) \wedge \mathcal{I}(\psi_{\theta,d-1})(x) \big).\end{aligned}$$

Observe that $\mathrm{qad}(\chi_{\alpha,\theta}) \leqslant \mathrm{qad}(\Psi) + 1$.

Now we obtain the desired sentence by defining $\varphi_\tau \;:=\; \bigvee_{(\alpha,\theta)\in R_\tau} (\xi_\alpha \wedge\; \chi_{\alpha,\theta})$.

Observe that, for some constant c depending only on σ, d, we have $|\xi_\alpha| \leqslant c$, $|\chi_{\alpha,\theta}| \leqslant c \cdot |\Psi| \cdot |\mathcal{T}_{\tilde{\sigma},q,d-1}|$, $|R_\tau| \leqslant c \cdot |\mathcal{T}_{\tilde{\sigma},q,d-1}|$, and $|\varphi_\tau| \leqslant c \cdot |\Psi| \cdot |\mathcal{T}_{\tilde{\sigma},q,d-1}|^2$. The claims about $|\Phi|$ and $\mathrm{qad}(\Phi)$ follow from the observations above. $\square$

Disconnected structures

We proceed with the preparations for the second step in the proof of our main theorem, where we lift the definability of q-order types from connected structures of tree-depth $\leqslant d$ to disconnected structures of tree-depth $\leqslant d$.

We consider each q-order type τ as a Boolean query on finite structures such that

$$() \in \tau(\mathfrak{A}) \iff \mathrm{tp}_q^<(\mathfrak{A}) = \tau.$$

For each structure $\mathfrak{A}$ and each Boolean query q, we let $n_q(\mathfrak{A})$ denote the number of components $\mathfrak{K}$ of $\mathfrak{A}$ such that $() \in q(\mathfrak{K})$. For each ordered set $Q := \{q_1, \ldots, q_\ell\}$ of Boolean queries, we let

$$\bar{n}_Q(\mathfrak{A}) := (n_{q_1}(\mathfrak{A}), \ldots, n_{q_\ell}(\mathfrak{A})).$$

For natural numbers $a, b, t \in \mathbb{N}^+$ we set

$$a \equiv_{\wedge t} b \quad \Leftrightarrow \quad (a = b \text{ or } a, b \geqslant t),$$

and we extend this relation to tuples $\bar{a}$ and $\bar{b}$ by saying $\bar{a} \equiv_{\wedge t} \bar{b}$ iff $a_i \equiv_{\wedge t} b_i$ for all components a_i and b_i.

We show that FO inherits its capability to count the types of components in q-ordered structures from its capability to distinguish linear orders of different length. The proof of the following lemma closely follows a step in the proof of [BS09b, Thm. 5.5]. Observe that for all $\mathfrak{A}, \mathfrak{B} \in \mathrm{FIN}_{\sigma,d}$, $n_{\mathcal{T}^{\mathrm{conn}}_{\sigma,q,d}}(\mathfrak{A}) \equiv_{\wedge t} n_{\mathcal{T}^{\mathrm{conn}}_{\sigma,q,d}}(\mathfrak{B})$ iff $n_{\mathcal{T}_{\sigma,q}}(\mathfrak{A}) \equiv_{\wedge t} n_{\mathcal{T}_{\sigma,q}}(\mathfrak{B})$.

Lemma 5.4.9. *Let $d \geqslant 1$, $q \in \mathbb{N}^+$ and $t := 2^q + 1$. Then for all $\mathfrak{A}, \mathfrak{B} \in \mathrm{FIN}_{\sigma,d}$,*

$$n_{\mathcal{T}_{\sigma,q}}(\mathfrak{A}) \equiv_{\wedge t} n_{\mathcal{T}_{\sigma,q}}(\mathfrak{B}) \implies \mathrm{tp}_q^<(\mathfrak{A}) = \mathrm{tp}_q^<(\mathfrak{B}).$$

Proof. For each component $\mathfrak{K}$ of $\mathfrak{A}$, we let $\preceq^{\mathfrak{K}}$ be a q-order of $\mathfrak{K}$. By part 2 of Definition 5.4.3, the q-orders on the components of $\mathfrak{A}$ can be extended to a q-order $\preceq^{\mathfrak{A}}$ on $\mathfrak{A}$ such that $\preceq^{\mathfrak{A}} \restriction \mathfrak{K} = \preceq^{\mathfrak{K}}$ for each component $\mathfrak{K}$ of $\mathfrak{A}$. We proceed analogously to obtain a q-order $\preceq^{\mathfrak{B}}$ on $\mathfrak{B}$. Let $\mathcal{T}_{\sigma,q} = \{\tau_1, \ldots, \tau_\ell\}$, where $\ell := |\mathcal{T}_{\sigma,q}|$ and $\tau_i \preceq_q \tau_j$ iff $i < j$. Consider the ordered word structures $(w_{\mathfrak{A}}, <)$, $(w_{\mathfrak{B}}, <)$ with $w_{\mathfrak{A}}, w_{\mathfrak{B}} \in \mathcal{T}^*_{\sigma,q}$ which we obtain from $(\mathfrak{A}, \preceq^{\mathfrak{A}})$ and $(\mathfrak{B}, \preceq^{\mathfrak{B}})$ by contracting each component $\mathfrak{K}$ to a single element that gets labelled by its q-type in the corresponding q-ordered structure. By this construction and by part 2 of Definition 5.4.3, we know that

$$w_{\mathfrak{A}} = \tau_1^{n_{\tau_1}(\mathfrak{A})} \cdots \tau_\ell^{n_{\tau_\ell}(\mathfrak{A})} \quad \text{and} \quad w_{\mathfrak{B}} = \tau_1^{n_{\tau_1}(\mathfrak{B})} \cdots \tau_\ell^{n_{\tau_\ell}(\mathfrak{B})}.$$

Since $n_{\mathcal{T}_{\sigma,q}}(\mathfrak{A}) \equiv_{\wedge t} n_{\mathcal{T}_{\sigma,q}}(\mathfrak{B})$, for each $i \in [1, \ell]$, either $n_{\tau_i}(\mathfrak{A}) = n_{\tau_i}(\mathfrak{B})$ or $n_{\tau_i}(\mathfrak{A}), n_{\tau_i}(\mathfrak{B}) \geqslant t$. A folklore result (cf. [Lib04, Ch. 3]) tells us that $(w_{\mathfrak{A}}, <) \equiv_q^{\mathrm{FO}} (w_{\mathfrak{B}}, <)$, i.e. Duplicator has a winning strategy in the q-round EF-game on the two ordered word structures.

We show that $(\mathfrak{A}, \preceq^{\mathfrak{A}}) \equiv_q^{\mathrm{FO}} (\mathfrak{B}, \preceq^{\mathfrak{B}})$. To this end, consider the following winning strategy for Duplicator in the q-round EF-game on $(\mathfrak{A}, \preceq^{\mathfrak{A}})$ and $(\mathfrak{B}, \preceq^{\mathfrak{B}})$. She maintains a *virtual* q-round EF-game $w_{\mathfrak{A}}$ on $w_{\mathfrak{B}}$ between a *Virtual Spoiler* and a *Virtual Duplicator*. When, during the i-th round, Spoiler chooses an element v in some component $\mathfrak{K}$ of, say, $\mathfrak{A}$, she lets the Virtual Spoiler play the corresponding position in $w_{\mathfrak{A}}$ in the i-th round of the virtual game. The Virtual Duplicator answers in $w_{\mathfrak{B}}$. Duplicator chooses a component $\mathfrak{K}'$ of $\mathfrak{B}$ for its reply according to the Virtual Duplicator's answer in $w_{\mathfrak{B}}$. The winning strategy on $(w_{\mathfrak{A}}, <)$ and $(w_{\mathfrak{B}}, <)$ tells us that $(\mathfrak{K}, \preceq^{\mathfrak{A}}) \equiv_q^{\mathrm{FO}} (\mathfrak{K}', \preceq^{\mathfrak{B}})$ and that all elements of $\mathfrak{K}$ and $\mathfrak{K}'$ have the same positions in $\preceq^{\mathfrak{A}}$ and $\preceq^{\mathfrak{B}}$ relative to the elements played in the previous rounds. Duplicator uses her winning strategy in the q-round game on the ordered components to determine the element of $\mathfrak{K}'$ that she uses as her answer to v. $\square$

For a tuple $\bar{a}$ of natural numbers, denote by $[\bar{a}]_{\wedge t}$ the tuple obtained from it by replacing all entries $> t$ with t. Then the previous lemma implies that if $\mathrm{td}(\mathfrak{A}) \leqslant d$, then $[\bar{n}_{\mathcal{T}^{\mathrm{conn}}_{\sigma,q,d}}(\mathfrak{A})]_{\wedge(2^q+1)}$ determines $\mathrm{tp}^<(\mathfrak{A})$. Hence, we obtain the following corollary:

Corollary 5.4.10. *Let $q, d \in \mathbb{N}^+$ and let $t := 2^q + 1$. For each $\varphi \in \mathrm{FO}[\sigma, <]$, let*

$$R_\varphi \ := \ \{[\bar{n}_{\mathcal{T}^{conn}_{\sigma,q,d}}(\mathfrak{A})]_{\wedge t} \mid \mathfrak{A} \in \mathrm{FIN}_{\sigma,d}, \mathrm{tp}_q^<(\mathfrak{A}) \models \varphi\}.$$

Then for each $\mathfrak{A} \in \mathrm{FIN}_{\sigma,d}$, we have

$$\mathrm{tp}_q^<(\mathfrak{A}) \models \varphi \quad \textit{iff} \quad [\bar{n}_{\mathcal{T}^{conn}_{\sigma,q,d}}]_{\wedge t} \in R_\tau.$$

Furthermore, $|\mathcal{T}_{\sigma,q,d}| \leqslant (t+1)^{|\mathcal{T}^{conn}_{\sigma,q,d}|}$.

The following lemma will be used in conjunction with the previous corollary to lift the definability of q-types from connected to disconnected structures.

Lemma 5.4.11. *Let $\mathrm{L} \in \{\mathrm{FO}, \mathrm{FO{+}MOD}\}$. For all $d, t \in \mathbb{N}^+$, every set of L-sentences Φ, and every set $R \subseteq [0, t]^{|\Phi|}$, there is an L-sentence ψ_R^Φ such that for each structure $\mathfrak{A}$ with $\mathrm{td}(\mathfrak{A}) \leqslant d$, we have*

$$\mathfrak{A} \models \psi_R^\Phi \iff [\bar{n}_\Phi(\mathfrak{A})]_{\wedge t} \in R.$$

Moreover, $|\psi_R^\Phi| \leqslant c \cdot |\Phi|^2 \cdot |R| \cdot t^2$ *and* $\mathrm{qad}(\psi_R^\Phi) \leqslant \mathrm{qad}(\Phi) + 2$*, for a constant* c *which depends only on* σ, d*.*

Proof. Let $\Phi := \{\varphi_1, \dots, \varphi_\ell\}$. Consider some $i \in [1,\ell]$ and let $\tilde{\varphi}_i(x) := \varphi_i \restriction \mathrm{reach}_d(x,z)$.

Let $n \in [1,t]$. We define a formula $\psi_i^n(\bar{x})$, where $\bar{x} := (x_1, \dots, x_n)$, which states that $x_1, \dots, x_n$ lie in distinct connected components, each of which satisfies φ_i:

$$\psi_i^n(\bar{x}) \ := \bigwedge_{j \in [1,n]} \tilde{\varphi}_i(x_j) \ \wedge \bigwedge_{j,k \in [1,n],\, j \neq k} \neg\, \mathrm{reach}_d(x_j, x_k).$$

Observe that $\mathrm{qad}(\psi_i^n) \leqslant \mathrm{qad}(\Phi)$ (in particular, since reach_d is an existential formula) and that $|\psi_i^n| \leqslant cn^2|\Phi| \leqslant ct^2|\Phi|$, for a constant c depending on σ, d only.

To obtain a formula which states that either the (pairwise disjoint) components of the $x_1, \dots, x_n$ are the only components which satisfy φ_i or the number of such components is at least t, we let

$$\psi_i^{n,t}(\bar{x}) \ := \ \begin{cases} \forall y \ \neg\tilde{\varphi}_i(y) & \text{if } n = 0, \\ \psi_i^n(\bar{x}) \wedge \forall y\, (\tilde{\varphi}_i(y) \rightarrow \bigvee_{i \in [1,n]} \mathrm{reach}_d(y, x_i)) & \text{if } 0 < n < t \\ \psi_i^n(\bar{x}) & \text{if } n \geqslant t. \end{cases}$$

Note that $\mathrm{qad}(\psi_i^{n,t}) \leqslant \mathrm{qad}(\Phi) + 1$ and $|\psi_i^{n,t}| \leqslant c \cdot |\psi_i^n|$, for some constant c depending on σ, d only. (Note that $|\psi_i^n| \geqslant n$, so the disjunction over $i \in [1,n]$ is absorbed by that.) We obtain the desired sentence $\psi_{R,t}^\Phi$ by setting

$$\psi_{R,t}^\Phi := \bigvee_{(n_1, \dots, n_\ell) \in R} \exists \bar{x}_i \bigwedge_{i \in [1,\ell]} \psi_i^{n_i,t}(\bar{x}_i),$$

where $\bar{x}_i$ is a tuple of n_i variables. Note that

$$\begin{aligned} |\psi_R^\Phi| \ &\leqslant \ |R| \cdot |\Phi| \cdot \max_{i \in [1,\ell]} |\psi_i^t| \ \leqslant \ c \cdot |R| \cdot |\Phi|^2 \cdot t^2, \\ \mathrm{qad}(\psi_R^\Phi) \ &\leqslant \ \max_{i \in [1,\ell]} \mathrm{qad}(\psi_i^{n_i,t}) + 1 \ \leqslant \ \mathrm{qad}(\Phi) + 2\,. \end{aligned}$$

□

Finally, we can prove our main theorem.

Proof of Theorem 5.4.1. By induction on the tree-depth d, we show that for each signature σ and each $\mathrm{FO}[\sigma,<]$-sentence φ with $\mathrm{qr}(\varphi) = q$, there is an $\mathrm{FO}[\sigma]$-sentence $\psi_{\varphi,d}$ with $|\psi_{\varphi,d}| \in \mathrm{EXP}_d(q)$ and $\mathrm{qad}(\psi_{\varphi,d}) \leqslant 3d$ such that for each $\mathfrak{A} \in \mathrm{FIN}_{\sigma,d}$, we have $\mathfrak{A} \models \psi_{\varphi,d}$ iff $\mathrm{tp}_q^<(\mathfrak{A}) \models \varphi$. Furthermore, we show that $|\mathcal{T}_{\sigma,q,d}| \in \mathrm{EXP}_d(q)$ and $|\mathcal{T}_{\sigma,q,d}^{\mathrm{conn}}| \in \mathrm{EXP}_{(d-1)}(q)$. To finish the proof, if φ is order-invariant, we let $\psi := \psi_{\varphi,d}$, and we obtain that $\mathfrak{A} \models_< \varphi$ iff $\mathfrak{A} \models \psi$.

Let $\mathcal{T}_{\sigma,q,d}^{\mathrm{conn}} = \{\theta_1, \dots, \theta_\ell\}$. First, for each $i \in [1,\ell]$, we construct a sentence φ_i that defines θ_i on $\mathrm{FIN}_{\sigma,d}^{\mathrm{conn}}$. If $d = 1$, observe that any *connected* structure $\mathfrak{A}$ of type $\theta_i \in \mathcal{T}_{\sigma,q,1}^{\mathrm{conn}}$ consists of a single element. The atomic σ-type α of this element determines the q-type of the unique q-order on $\mathfrak{A}$. The FO-sentence $\varphi^{\text{conn}}_{\tau,1} := \exists x\, \alpha(x)$ hence defines τ on $\text{FIN}^{\text{conn}}_{\sigma,1}$. We obviously have $|\varphi^{\text{conn}}_{\tau,1}| \leqslant c \cdot |\sigma|$, for some absolute constant c, and $|\mathcal{T}^{\text{conn}}_{\sigma,q,d}| \leqslant 2^{|\sigma|} \in \text{EXP}_{(d-1)}(q)$.

If $d > 1$, we construct an FO-sentence $\psi_{\theta,d-1}$ inductively for each q-type $\theta \in \mathcal{T}_{\tilde{\sigma},q,d-1}$. Let $\Psi := \{\psi_{\theta,d-1} : \theta \in \mathcal{T}_{\tilde{\sigma},q,d-1}\}$. By induction, we obtain $|\Psi| \in \text{EXP}_{(d-1)}(q)$, and $\text{qad}(\Psi) \leqslant 3(d-1)$, and we have $|\mathcal{T}_{\tilde{\sigma},q,d-1}| \in \text{EXP}_{(d-1)}(q)$. We construct φ_i according to Lemma 5.4.8, i.e. we let $\varphi_i := \varphi^{\text{conn}}_{\theta_i,d}$ for each $i \leqslant \ell$. Let $\Phi := \{\varphi_1, \ldots, \varphi_\ell\}$. Then there is a constant c depending only on σ, d, such that

$$\begin{aligned} |\Phi| \leqslant c \cdot |\Psi| \cdot |\mathcal{T}_{\tilde{\sigma},q,d-1}|^2 \ &\in \ \text{EXP}_{(d-1)}(q) \quad \text{and} \\ \text{qad}(\Phi) \leqslant \text{qad}(\Psi) + 2 \ &\leqslant \ 3(d-1) + 2. \end{aligned}$$

Now consider a sentence $\varphi \in \text{FO}[\sigma, <]$. Let $R := R_\varphi$ be given by Corollary 5.4.10. We apply Lemma 5.4.11 with $t := 2^q + 1$ to obtain a sentence $\psi_{\varphi,d} := \psi^{\Phi}_{R}$. To see that $\psi_{\varphi,d}$ is defined correctly, consider some $\mathfrak{A} \in \text{FIN}_{\sigma,d}$. Observe that for each $i \in [1, \ell]$ and each component $\mathfrak{K}$ of $\mathfrak{A}$, we have $\mathfrak{K} \models \varphi_i$ iff $\text{tp}^<_q(\mathfrak{K}) = \tau_i$, and thus $\bar{n}_\Phi(\mathfrak{A}) = \bar{n}_{\mathcal{T}^{\text{conn}}_{\sigma,q,d}}(\mathfrak{A})$. Then

$$\begin{aligned} \mathfrak{A} \models \psi_{\varphi,d} \text{ iff } & [\bar{n}_{\mathcal{T}^{\text{conn}}_{\sigma,q,d}}(\mathfrak{A})]_{\wedge t} \in R && \text{(by Lemma 5.4.11 and the previous observation)} \\ \text{iff } & \text{tp}^<_q(\mathfrak{A}) \models \varphi. && \text{(by Corollary 5.4.10)} \end{aligned}$$

By Lemma 5.4.11, for some constant c depending only on σ, d, we have

$$\begin{aligned} |\psi_{\varphi,d}| &\leqslant c \cdot |\Phi|^2 \cdot |R| \cdot t^2 \quad \text{and} \\ \text{qad}(\psi_{\varphi,d}) &\leqslant \ \text{qad}(\Phi) + 1 \ \leqslant \ 3d\,. \end{aligned}$$

Observe that $|\Phi| = \ell = |\mathcal{T}^{\text{conn}}_{\sigma,q,d}| \in \text{EXP}_{(d-1)}(q)$ by Corollary 5.4.6 and that $|R| \leqslant t^\ell \in \text{EXP}_d(q)$. Hence, $|\psi_{\varphi,d}| \in \text{EXP}_d(q)$. By Corollary 5.4.10, we also obtain $|\mathcal{T}_{\sigma,q,d}| \in \text{EXP}_d(q)$. □

5.5 Order-invariant MSO

Courcelle [Cou96, Thm. 4.1] proved that classes of graphs definable by order-invariant MSO sentences are recognisable.[10] He conjectured [Cou91, Conjecture1] that all recognisable sets of graphs of *bounded tree-width* are definable in MSO with modulo-counting (CMSO). This conjecture was open for a very long time and has seen several flawed proof attempts (cf. [CE12, p. 574]) published. Fortunately, it seems that it has been resolved very recently [BP16]. It is known that this implies that $<$-inv-MSO is equivalent to CMSO on graphs of bounded tree-width. Remarkably, this has been proved independently and the paper [EFG16] which proves this, among other other results, has been published in the proceedings of the same conference as [BP16]. The difficult part in this result is the construction of an CMSO-sentence for a given $<$-inv-MSO-sentence. It is well-known and easy to see that, on ordered structure, all modulo-counting quantifiers can be defined in MSO. That is, regardless of the considered class of structures, for each sentence of

[10] See [Cou96] for the precise definition of the notion of recognisability, which is not important here.

modulo-counting MSO there is an equivalent $<$-inv-MSO-sentence. We show that in the special case of structures of bounded tree-depth, $<$-inv-MSO collapses to FO+MOD, i.e. *first-order* logic with modulo-counting.

Theorem 5.5.1. *For every $d \in \mathbb{N}^+$ and every $<$-inv-MSO-sentence φ there is an* FO+MOD-*sentence ψ with $\mathrm{qad}(\psi) \leqslant 3d$ which is equivalent to φ on $\mathrm{FIN}_{\sigma,d}$.*

In contrast to the previous sections, we do not analyse the formula size. It is known from [GS05] that MSO, even without modulo-counting, can define the length of orders non-elementarily more succinct than FO. Using the MSO-sentence of [GS05] and the Ehrenfeucht-Fraïssé-game for FO+MOD (see, e.g., [Nur96]), it can be easily shown that $<$-inv-MSO is already non-elementarily more succinct than FO+MOD on structures with empty relations. Note that, to this end, it is also important that our definition of the length of a FO+MOD-formula takes the size of the numbers used in the modulo-counting quantifiers into account.

For the proof of Theorem 5.5.1, we proceed similarly to the last section. Again we need to understand $<$-inv-MSO's capabilities to count the number of components of a given q-type in q-ordered structures. However, this time we need to count not only up to some threshold, but also modulo some fixed divisor.

Recall that, for $n \in \mathbb{N}$, $p \in \mathbb{N}^+$, we write $n \operatorname{rem} p$ for the remainder of the division of n by p. We extend this to tuples $\bar{n} := (n_1, \ldots, n_\ell) \in \mathbb{N}^\ell$, by defining

$$\bar{n} \operatorname{rem} p := (n_1 \operatorname{rem} p, \ldots, n_\ell \operatorname{rem} p).$$

We write $\bar{n} \equiv_{\mathrm{mod}\, p} \bar{m}$ iff $\bar{n} \operatorname{rem} p = \bar{m} \operatorname{rem} p$.

Below, we prove the following lemma which shows that MSO inherits its component counting capabilities in q-ordered structures from its capabilities to distinguish orders of different lengths.

Lemma 5.5.2. *For each $q \in \mathbb{N}^+$, there is a $p \in \mathbb{N}^+$ such that for all q-ordered structures $(\mathfrak{A}, \preceq^{\mathfrak{A}})$ and $(\mathfrak{B}, \preceq^{\mathfrak{B}})$,*

$$\text{if } \bar{n}_{\mathcal{T}_{\sigma,q}}(\mathfrak{A}) \equiv_{\mathrm{mod}\, p} \bar{n}_{\mathcal{T}_{\sigma,q}}(\mathfrak{B}) \text{ and } \bar{n}_{\mathcal{T}_{\sigma,q}}(\mathfrak{A}) \equiv_{\wedge p} \bar{n}_{\mathcal{T}_{\sigma,q}}(\mathfrak{B}) \quad \text{then} \quad (\mathfrak{A}, \preceq^{\mathfrak{A}}) \equiv_q^{\mathrm{MSO}} (\mathfrak{B}, \preceq^{\mathfrak{B}}).$$

In the following, we say that an ordered structure $(\mathfrak{A}, \preceq)$ is *component ordered*, if the order $\preceq$ is a sum of the orders on the components of $\mathfrak{A}$, i.e. for some enumeration $\mathfrak{K}_1, \ldots, \mathfrak{K}_n$ of the components of $\mathfrak{A}$, we have $\preceq = \preceq\restriction K_1 + \preceq\restriction K_2 + \cdots + \preceq\restriction K_n$. Observe that q-ordered structures are also component ordered. It will be convenient to have some notation that allows us to treat component ordered structures similarly to words. Given two ordered structures $(\mathfrak{A}, \preceq^{\mathfrak{A}})$ and $(\mathfrak{B}, \preceq^{\mathfrak{B}})$, we let $(\mathfrak{A}, \preceq^{\mathfrak{A}}) \sqcup (\mathfrak{B}, \preceq^{\mathfrak{B}}) := (\mathfrak{A} \sqcup \mathfrak{B}, \preceq^{\mathfrak{A}} + \preceq^{\mathfrak{B}})$, where $\mathfrak{A} \sqcup \mathfrak{B}$ denotes the disjoint union of $\mathfrak{A}$ and $\mathfrak{B}$ and we consider $\preceq^{\mathfrak{A}}, \preceq^{\mathfrak{B}}$ as orders on the components of the disjoint union (via the inclusion mappings for $\mathfrak{A}, \mathfrak{B}$). Instead of $(\mathfrak{A}, \preceq^{\mathfrak{A}}) \sqcup (\mathfrak{B}, \preceq^{\mathfrak{B}})$, we also write $(\mathfrak{A}, \preceq^{\mathfrak{A}})(\mathfrak{B}, \preceq^{\mathfrak{B}})$. Like in the following definition, we often omit the order to make this notation less cluttered. For each component ordered structure $\mathfrak{A}$, we define its *i-th power* $\mathfrak{A}^i$ by $\mathfrak{A}^1 := \mathfrak{A}$ and $\mathfrak{A}^i := \mathfrak{A}^{i-1}\mathfrak{A}$ if $i > 1$.

The proof of Lemma 5.5.2 rests on the following lemma.

Lemma 5.5.3 (Pumping Lemma). *For each $q \in \mathbb{N}^+$, there is a number $p \in \mathbb{N}^+$ such that for all component ordered structures $\mathfrak{A}$ and all $r \in \mathbb{N}$, $i, j \in \mathbb{N}^+$,*

$$\mathfrak{A}^{r+ip} \equiv_q^{\mathrm{MSO}} \mathfrak{A}^{r+jp}.$$

Proof. Let $\mathcal{T}$ denote the (finite) set of q-types which are realised by component ordered σ-structures. We lift the disjoint union of ordered structures to $\mathcal{T}$ by defining $\mathrm{tp}_q(\mathfrak{A}) \sqcup \mathrm{tp}_q(\mathfrak{B}) := \mathrm{tp}_q(\mathfrak{A} \sqcup \mathfrak{B})$. The Composition Lemma (Lemma 5.4.5) shows that this operation is well-defined. It is also associative, so that $(\mathcal{T}, \sqcup)$ is a finite semigroup. Hence, there is a number p such that for each $\tau \in \mathcal{T}$, τ^p is idempotent (cf. e.g. [How76]), i.e. $\tau^p = \tau^{ip}$ for each $i \in \mathbb{N}^+$. Then, for all $\mathfrak{A}, r, i, p$ as in the statement of the lemma, $\mathrm{tp}_q(\mathfrak{A})^{r+ip} = \mathrm{tp}_q(\mathfrak{A})^{r+jp}$, i.e. $\mathfrak{A}^{r+ip} \equiv_q^{\mathrm{MSO}} \mathfrak{B}^{r+jp}$. □

Proof of Lemma 5.5.2. Let $\mathcal{T}_{\sigma,q} = \{\tau_1, \dots, \tau_\ell\}$ with $\tau_i \prec_q \tau_j$ iff $i < j$. For each $i \in [1, \ell]$, fix a connected q-ordered structure $\mathfrak{K}_i$ whose type is $\mathrm{tp}_q(\mathfrak{K}_i) = \tau_i$. By repeated application of the Composition Lemma, we can assume without loss of generality that $\mathfrak{K} \cong \mathfrak{K}_i$ for each q-ordered component $\mathfrak{K}$ of $\mathfrak{A}$ or $\mathfrak{B}$ with $\mathrm{tp}_q(\mathfrak{K}) = \tau_i$. Let $n_i := n_{\tau_i}(\mathfrak{A})$ and let $m_i := n_{\tau_i}(\mathfrak{B})$ for each $i \in [1, \ell]$. By part 2 of Definition 5.4.3, we obtain

$$\mathfrak{A} \cong \mathfrak{K}_1^{n_1} \mathfrak{K}_2^{n_2} \cdots \mathfrak{K}_\ell^{n_\ell} \quad \text{and} \quad \mathfrak{B} \cong \mathfrak{K}_1^{m_1} \mathfrak{K}_2^{m_2} \cdots \mathfrak{K}_\ell^{m_\ell}.$$

For each $i \in [1, \ell]$, we have $n_{\tau_i}(\mathfrak{A}) \equiv n_{\tau_i}(\mathfrak{B}) \,(\mathrm{mod}\, p)$, i.e. there are $r_i \in [p]$ and $a_i, b_i \in \mathbb{N}$ such that $n_i = r_i + a_i p$ and $m_i = r_i + b_i p$. Furthermore, as $n_{\tau_i}(\mathfrak{A}) \equiv_{\wedge p} n_{\tau_i}(\mathfrak{B})$, we have $a_i > 0$ iff $b_i > 0$. By repeated application of the Pumping Lemma, we obtain

$$\mathfrak{K}_1^{n_1} \mathfrak{K}_2^{n_2} \cdots \mathfrak{K}_\ell^{n_\ell} \;\equiv_q^{\mathrm{MSO}}\; \mathfrak{K}_1^{r_1+b_1p} \mathfrak{K}_2^{r_2+b_2p} \cdots \mathfrak{K}_\ell^{r_\ell+b_\ell p} \;=\; \mathfrak{K}_1^{m_1} \mathfrak{K}_2^{m_2} \cdots \mathfrak{K}_\ell^{m_\ell}.$$

Hence, $\mathfrak{A} \equiv_q^{\mathrm{MSO}} \mathfrak{B}$. □

The next lemma is a modulo-counting analogue of Lemma 5.4.11.

Lemma 5.5.4. *For all $d, p \in \mathbb{N}^+$, each set of FO+MOD$[\sigma]$-sentences Φ, and each set $R \subseteq [p{+}1]^\ell \times [p]^\ell$, there is an FO+MOD$[\sigma]$-sentence χ_R^Φ such that for each $\mathfrak{A} \in \mathrm{FIN}_{\sigma,d}$,*

$$\mathfrak{A} \models \chi_R^\Phi \quad \textit{iff} \quad ([\bar{n}_\Phi(\mathfrak{A})]_{\wedge p}, \bar{n}_\Phi(\mathfrak{A}) \,\mathrm{rem}\, p) \in R.$$

Furthermore, $\mathrm{qad}(\chi_R^\Phi) \leqslant \max\{\mathrm{qad}(\Phi) + 2, 2(d-1) + 1\}$.

In contrast to Lemma 5.4.11, the proof of Lemma 5.5.4 is not straightforward, because it is not obvious how modulo-counting quantifiers can be used to count the number of components satisfying a given FO+MOD-sentence. Remedy to this problem is provided by Lemma 5.3.3 which shows that the number of tree-depth roots of each component of a structure can be bounded in terms of its tree-depth only.

Proof of Lemma 5.5.4. For each $r \in [p]$ and $\varphi \in$ FO+MOD$[\sigma]$, we construct a sentence χ_r^φ such that $\mathfrak{A} \models \chi_r^\varphi$ iff $n_\varphi(\mathfrak{A}) \equiv r \,(\mathrm{mod}\, p)$, i.e. the number of components in $\mathfrak{A}$ with $\mathfrak{A} \models \varphi$ is congruent to r modulo p. The formula we construct will have $\mathrm{qad}(\chi_r^\varphi) \leqslant \max\{\mathrm{qad}(\varphi) +$

$1, 2(d-1)+2\}$. Using the formulae $\varphi^{\Phi}_{\{\bar{n}\}}$ given by Lemma 5.4.11 for $t = p$ we can then define

$$\chi^{\Phi}_{R} := \bigvee_{(\bar{n},\bar{r})\in R} \Big(\varphi^{\Phi}_{\{\bar{n}\}} \wedge \bigwedge_{i\in[\ell]} \chi^{r_i}_{\varphi_i}\Big).$$

Obviously, $\mathrm{qad}(\chi^{\Phi}_{R}) \leqslant \max\{\mathrm{qad}(\Phi)+2, 2(d-1)+2\}$.

Thus consider some $r \in [p]$ and $\varphi \in$ FO+MOD$[\sigma]$. We define a formula $\varphi^{=k}(x)$, such that $\mathfrak{A} \models \varphi^{=k}(a)$, for $\mathfrak{A} \in \mathrm{FIN}_{\sigma,d}$ and $a \in A$, iff a is a root of a component $\mathfrak{K}$ of $\mathfrak{A}$ which has k roots in total and such that $\mathfrak{K} \models \varphi$, $a \in \mathrm{roots}(\mathfrak{K})$:

$$\begin{aligned}\varphi^{=k}(x) := \varphi\restriction\mathrm{reach}_d(x,z) \ \wedge\ \tilde{\mathrm{roots}}_d(x)&\\ \wedge\ \exists x_1 \ldots \exists x_k \ \Big(\bigwedge_{j\in[1,k]} \big(\tilde{\mathrm{roots}}_d(x_j) \wedge \mathrm{reach}_d(x_j,x)\ \wedge \bigwedge_{j,j'\in[1,k],\, j\neq j'} x_j \neq x_{j'}\big)&\\ \wedge\ \forall y\ \big(\tilde{\mathrm{roots}}_d(y) \wedge \bigwedge_{j\in[1,k]} y \neq x_j\big) \to \bigwedge_{j\in[1,k]} \neg\,\mathrm{reach}_d(y,x)\Big),&\end{aligned}$$

where $\tilde{\mathrm{roots}}_d(x) := \mathrm{roots}_d(x)\restriction\mathrm{reach}_d(x,z)(x)$. Observe that

$$\begin{aligned}\mathrm{qad}(\varphi^{=k}) &\leqslant \max\{\mathrm{qad}(\tilde{\varphi}), \mathrm{qad}(\tilde{\mathrm{roots}}_d)+1, \mathrm{qad}(\mathrm{reach}_d)+1\}\\ &\leqslant \max\{\mathrm{qad}(\varphi), 2(d-1)+1\}\end{aligned}$$

Let $b := f(d)$ be such that $|\mathrm{roots}(G)| \leqslant b$ for all connected graphs of tree-depth at most d (cf. Lemma 5.3.3). For a structure $\mathfrak{A} \in \mathrm{FIN}_{\sigma,d}$ let H be the set of components which satisfy φ. We partition H into sets $H_1, \ldots, H_b$ such that H_k contains exactly those components which satisfy φ and have exactly k roots. The formula χ^{φ}_{r} shall be such that $\mathfrak{A} \models \chi^{\varphi}_{r}$ iff $|H| \equiv r \pmod{p}$.

Note that the number of elements $a \in A$ such that $\mathfrak{A} \models \varphi^{=k}(a)$ is $k \cdot |H_k|$. Let $M \subseteq [p]^b$ be the set of all $(a_1, \ldots, a_b)$ such that $\sum_{k\in[1,b]} a_k \equiv r \pmod{p}$. We set

$$\chi^{\varphi}_{r} := \bigvee_{(a_1,\ldots,a_b)\in M} \ \bigwedge_{k\in[1,b]} \exists^{k\cdot a_k \ (\mathrm{mod}\ k\cdot p)} x\, \varphi^{=k}(x).$$

Then $\mathrm{qad}(\chi^{\varphi}_{r}) \leqslant \max\{\mathrm{qad}(\varphi), 2(d-1)+1\}+1$.

We show that indeed $\mathfrak{A} \models \chi^{\varphi}_{r}$ iff $n_\varphi \equiv r \pmod{p}$. For the "if" part, assume that $n_\varphi(\mathfrak{A}) \equiv r \pmod{p}$. By our definition of H and the H_k we get $n_\varphi(\mathfrak{A}) = |H| = \sum_{k\in[1,b]} |H_k| \equiv r \pmod{p}$. Write $|H_k| = b_k p + a_k$ with $a_k \in [0, p-1]$. Then $\sum_{k\in[1,b]} a_k \equiv r \pmod{p}$, so $(a_1, \ldots, a_b) \in M$. On the other hand,

$$k \cdot |H_k| = k(b_k p + a_k) = pkb_k + k \cdot a_k$$

with $k \cdot a_k \in [0, k \cdot p]$, so $\mathfrak{A} \models \exists^{k\cdot a_k \ (\mathrm{mod}\ k\cdot p)} x\, \varphi^{=k}(x)$ for all $k \in [1, b]$. Thus $\mathfrak{A} \models \chi^{\varphi}_{r}$.

For the converse direction, assume $\mathfrak{A} \models \chi^{\varphi}_{r}$. Then

$$\mathfrak{A} \models \bigwedge_{k\in[1,b]} \exists^{k\cdot a_k \ (\mathrm{mod}\ k\cdot p)} x\, \varphi^{=k}(x)$$

for some $(a_1, \dots, a_b)$ with $\sum_{k \in [1,b]} a_k \equiv r \pmod{p}$. Therefore $k \cdot |H_k| = k \cdot p \cdot b_k + k \cdot a_k = k(p \cdot b_k + a_k)$ for some $b_k \in \mathbb{N}$, and $|H_k| \equiv a_k \pmod{p}$. In particular $n_\varphi(\mathfrak{A}) = \sum_{k \in [1,b]} |H_k| = \sum_{k \in [1,b]} a_k \equiv r \pmod{p}$. $\square$

With these preparations, the proof of Theorem 5.5.1 is very similar to the proof of Theorem 5.4.1.

Proof of Theorem 5.5.1. The proof proceeds by induction on the tree-depth d. We show that for each MSO$[\sigma, <]$-sentence φ with $\mathrm{qr}(\varphi) = q$, there is an FO+MOD$[\sigma]$-sentence $\psi_{\varphi,d}$ such that for each $\mathfrak{A} \in \mathrm{FIN}_{\sigma,d}$, we have $\mathfrak{A} \models \psi_{\varphi,d}$ iff $\mathrm{tp}_q^<(\mathfrak{A}) \models \varphi$. In particular, if φ is order-invariant, we let $\psi := \psi_{\varphi,d}$, and we obtain $\mathfrak{A} \models_< \varphi$ iff $\mathfrak{A} \models \psi := \psi_{\varphi,d}$.

Let $\mathcal{T}^{\mathrm{conn}}_{\sigma,q,d} = \{\theta_1, \dots, \theta_\ell\}$. We construct a sentence φ_i that defines θ_i on $\mathrm{FIN}^{\mathrm{conn}}_{\sigma,d}$, for each $i \in [1, \ell]$. If $d = 1$, the type of a connected structure of type θ_i is determined by the atomic σ-type α of its single element. We let $\varphi^{\mathrm{conn}}_{\tau,1} := \exists x\, \alpha(x)$. If $d > 1$, for each q-type $\theta \in \mathcal{T}_{\tilde{\sigma},q,d-1}$, we obtain an FO+MOD-sentence $\psi_{\theta,d-1}$ with $\mathrm{qad}(\psi_{\theta,d-1}) \leqslant 3(d-1)$.

We construct φ_i according to Lemma 5.4.8, i.e. we let $\varphi_i := \psi^{\mathrm{conn}}_{\theta_i,d}$ for each $i \leqslant \ell$. Let $\Phi := \{\varphi_1, \dots, \varphi_\ell\}$. Note that $\mathrm{qad}(\Phi) \leqslant 3(d-1) + 2$.

Now consider a sentence $\varphi \in \mathrm{MSO}[\sigma, <]$. Let

$$R := \left\{ \left([\bar{n}_{\mathcal{T}_{\sigma,q}}(\mathfrak{B})]_{\wedge p}, \bar{n}_{\mathcal{T}_{\sigma,q}}(\mathfrak{B}) \operatorname{rem} p \right) \mid \mathfrak{B} \in \mathrm{FIN}_{\sigma,d}, \mathrm{tp}_q^<(\mathfrak{B}) \models \varphi \right\}$$

where p is given by the Pumping Lemma for q. We construct $\psi_{\varphi,d} := \psi_R^\Phi$ according to Lemma 5.5.4. In particular, $\mathrm{qad}(\psi_{\varphi,d}) \leqslant \mathrm{qad}(\Phi) + 1 \leqslant 3d$. Consider some $\mathfrak{A} \in \mathrm{FIN}_{\sigma,d}$. Observe that, for each component $\mathfrak{K}$ of $\mathfrak{A}$, we have $\mathfrak{K} \models \varphi_i$ iff $\mathrm{tp}_q^<(\mathfrak{K}) = \tau_i$. Hence, $([\bar{n}_\Phi(\mathfrak{A})]_{\wedge p}, \bar{n}_\Phi(\mathfrak{A}) \operatorname{rem} p) = ([\bar{n}_{\mathcal{T}_{\sigma,q}}(\mathfrak{A})]_{\wedge p}, \bar{n}_{\mathcal{T}_{\sigma,q}}(\mathfrak{A}) \operatorname{rem} p)$. Thus

$$\mathfrak{A} \models \psi_{\varphi,d} \iff ([\bar{n}_{\mathcal{T}_{\sigma,q}}(\mathfrak{A})]_{\wedge p}, \bar{n}_{\mathcal{T}_{\sigma,q}}(\mathfrak{A}) \operatorname{rem} p) = ([\bar{n}_{\mathcal{T}_{\sigma,q}}(\mathfrak{B})]_{\wedge p}, \bar{n}_{\mathcal{T}_{\sigma,q}}(\mathfrak{B}) \operatorname{rem} p)$$

for some structure $\mathfrak{B} \in \mathrm{FIN}_{\sigma,d}$ with $\mathrm{tp}_q^<(\mathfrak{B}) \models \varphi$. As a consequence of Lemma 5.5.2, this holds iff $\mathrm{tp}_q^<(\mathfrak{A}) \models \varphi$. $\square$

5.6 MSO

In [EGT12] it was proved that each MSO-definable class of finite graphs of bounded tree-depth is also FO-definable. Our approach towards the results of the previous section can be adapted to obtain another proof of this result which allows us to give an elementary upper bound on the size of the FO-sentence in terms of the quantifier-rank of the MSO-sentence. Throughout this section, we assume in all notation whose definition refers to a logic L that L = MSO. We let

$$\mathcal{T}_{\sigma,q,d} := \{\mathrm{tp}_q^{\mathrm{MSO}}(\mathfrak{A}) : \mathfrak{A} \in \mathrm{FIN}_{\sigma,d}\}$$

and we let

$$\mathcal{T}^{\mathrm{conn}}_{\sigma,q,d} := \{\mathrm{tp}_q^{\mathrm{MSO}}(\mathfrak{A}) : \mathfrak{A} \in \mathrm{FIN}^{\mathrm{conn}}_{\sigma,d}\}.$$

Theorem 5.6.1. *Let $d \in \mathbb{N}^+$ and let σ be a signature. For each* $\mathrm{MSO}[\sigma]$*-sentence φ there is an* $\mathrm{FO}[\sigma]$*-sentence ψ with $|\psi| \in \mathrm{EXP}_d(\mathrm{qr}(\varphi))$ and $\mathrm{qad}(\psi) \leqslant 3d$ that is equivalent to φ on* $\mathrm{FIN}_{\sigma,d}$.

In Section 5.7.1, we prove a lower bound which shows that the dependence on d in the upper bound on $|\psi|$ is essentially optimal.

Much of the proof of Theorem 5.6.1 follows the proof of Theorem 5.4.1, but we are spared of the complications that arose in connection with the ordering of structures. Overall, this makes the proof of Theorem 5.6.1 simpler. On the other hand, the proof of an analogue to Lemma 5.4.9 becomes somewhat more complicated.

5.6.1 Counting components

In Lemma 5.4.9, we did not use the fact that we consider only structures of bounded tree-depth. Here naively ignoring the bounded tree-depth would cause the component counting threshold for MSO-sentences of quantifier-rank q to depend non-elementarily on q. We use the following lemma to avoid this.

Lemma 5.6.2. *Let $d, q \in \mathbb{N}^+$. There is a $t := t(d,q) \in \mathrm{EXP}_d(q)$ such that for all structures $\mathfrak{A}, \mathfrak{B} \in \mathrm{FIN}_{\sigma,d}$,*

$$\bar{n}_{\mathcal{T}_{\sigma,q}}(\mathfrak{A}) \equiv_{\wedge t} \bar{n}_{\mathcal{T}_{\sigma,q}}(\mathfrak{B}) \quad \Longrightarrow \quad \mathfrak{A} \equiv_q^{\mathrm{MSO}} \mathfrak{B}.$$

Lemma 5.6.2 is an easy consequence of the following two lemmas.

Lemma 5.6.3. *Let $k \in \mathbb{N}^+$, $q \in \mathbb{N}$, and $t := 2^{kq}$. Let σ be a signature. For all structures $\mathfrak{A}, \mathfrak{B} \in \mathrm{FIN}_\sigma$ whose components each contain at most k elements,*

$$\bar{n}_{\mathcal{T}_{\sigma,q}}(\mathfrak{A}) \equiv_{\wedge t} \bar{n}_{\mathcal{T}_{\sigma,q}}(\mathfrak{B}) \quad \Longrightarrow \quad \mathfrak{A} \equiv_q^{\mathrm{MSO}} \mathfrak{B}.$$

Lemma 5.6.4. *Let $d, q \in \mathbb{N}^+$ and let σ be a signature. Each structure $\mathfrak{A} \in \mathrm{FIN}_{\sigma,d}$ contains an induced substructure $\mathfrak{B}$ with $|B| \in \mathrm{EXP}_d(q)$ and $\mathfrak{A} \equiv_q^{\mathrm{MSO}} \mathfrak{B}$. If $\mathfrak{A}$ is connected, there is such a structure $\mathfrak{B}$ with $|B| \in \mathrm{EXP}_{(d-1)}(q)$.*

Before we prove Lemma 5.6.3 and Lemma 5.6.4, we show how to prove Lemma 5.6.2 with their help. The proof will also use the following variant of a standard composition lemma, which we take for granted (we use a variant for signatures with constants, where the constant symbols will be used in the proof of Lemma 5.6.3).

The definition of the disjoint union $\mathfrak{A} \sqcup \mathfrak{B}$ of structures $\mathfrak{A}$ and $\mathfrak{B}$ can be extended to signatures with constant symbols, if the constant symbols of $\mathfrak{A}$ and $\mathfrak{B}$ are disjoint.

Lemma 5.6.5 (Composition Lemma). *Let $q \in \mathbb{N}$. Let σ_1, σ_2 be signatures which may contain constant symbols, where the constants in σ_1 and σ_2 are disjoint. If $\mathfrak{A}_1, \mathfrak{B}_1$ are σ_1-structures and $\mathfrak{A}_2, \mathfrak{B}_2$ are σ_2-structures such that $\mathfrak{A}_1 \equiv_q^{\mathrm{MSO}} \mathfrak{B}_1$ and $\mathfrak{A}_2 \equiv_q^{\mathrm{MSO}} \mathfrak{B}_2$, then*

$$\mathfrak{A}_1 \sqcup \mathfrak{A}_2 \equiv_q^{\mathrm{MSO}} \mathfrak{B}_1 \sqcup \mathfrak{B}_2.$$

Proof of Lemma 5.6.2. With the help of Lemma 5.6.4 and the Composition Lemma, we can assume without loss of generality that $\mathfrak{A}$ and $\mathfrak{B}$ contain only components of size at most $k \in \text{EXP}_{(d-1)}(q)$. Let $t := 2^{kq}$ as in Lemma 5.6.3. Then $t \in \text{EXP}_d(q)$ and hence the claim follows from Lemma 5.6.3. $\square$

Proof of Lemma 5.6.3. For the proof, we consider signatures σ which contain constant symbols. In this case, the components of a σ-structure are not necessarily σ-structures, because they might not contain all constants. Let $T_{\sigma,q}$ denote the union of the sets of (MSO, σ', q)-types over all signatures $\sigma' \subseteq \sigma$. For σ-structures $\mathfrak{A}, \mathfrak{B}$ and $q, t \in \mathbb{N}^+$, we write $\mathfrak{A} \approx_{q,t} \mathfrak{B}$ if $\bar{n}_{T_{\sigma,q}} \equiv_{\wedge t} \bar{n}_{T_{\sigma,q}}$.

By induction on q, we prove the stronger claim that for each signature σ *which may contain constant symbols* and all σ-structures $\mathfrak{A}$ and $\mathfrak{B}$ whose components each contain at most k elements,

$$\mathfrak{A} \approx_{q,t} \mathfrak{B} \implies \mathfrak{A} \equiv_q^{\text{MSO}} \mathfrak{B}.$$

Let $q = 0$. Since $\mathfrak{A} \approx_{q,1} \mathfrak{B}$, there exists a bijection f between the sets $M_{\mathfrak{A}}, M_{\mathfrak{B}}$ of components of $\mathfrak{A}, \mathfrak{B}$ which contain constants. Furthermore, this bijection preserves the 0-type of components, i.e. for each component $\mathfrak{K} \in M_{\mathfrak{A}}$ there exists a partial isomorphism $g_{\mathfrak{K}}$ whose domain and codomain are, respectively, the set of constants of $\mathfrak{K}$ and $f(\mathfrak{K})$. These partial isomorphisms can be extended to a partial isomorphism $g := \bigcup_{\mathfrak{K} \in M_{\mathfrak{A}}} g_{\mathfrak{K}}$ of $\mathfrak{A}$ and $\mathfrak{B}$ whose domain and codomain are, respectively, the set of constants of $\mathfrak{A}$ and $\mathfrak{B}$. Hence, $\mathfrak{A} \equiv_0^{\text{MSO}} \mathfrak{B}$.

For each $q \in \mathbb{N}$, let $t(q) := 2^{kq}$. Now let $q > 0$. We consider the case where $\mathfrak{A}$ and $\mathfrak{B}$ contain only components of a single q-type τ over some signature $\sigma' \subseteq \sigma$. The general case follows by an application of the Composition Lemma. By a further application of the Composition Lemma, we can assume that all components of $\mathfrak{A}$ and $\mathfrak{B}$ are isomorphic to a single structure $\mathfrak{K}$ of type τ. Now if $n_\tau(\mathfrak{A}) = n_\tau(\mathfrak{B})$, then $\mathfrak{A}$ and $\mathfrak{B}$ are isomorphic, so we are done. Assume that $n_\tau(\mathfrak{A}), n_\tau(\mathfrak{B}) > t(q)$. We show that Duplicator wins the q-round EF-game on $\mathfrak{A}$ and $\mathfrak{B}$.

Consider the first round of the game. Suppose that Spoiler plays a point move, i.e. he chooses an element, say, $a \in A$. Duplicator chooses an element b corresponding to a in a copy of $\mathfrak{K}$ in $\mathfrak{B}$. This introduces exactly one component of a new isomorphism-type τ' in each of $(\mathfrak{A}, a)$ and $(\mathfrak{B}, b)$. The remaining components of $(\mathfrak{A}, a)$, $(\mathfrak{B}, b)$ all remain their isomorphism type and there are more than $t(q) - 1 \geqslant t(q-1)$ such components. Hence, $(\mathfrak{A}, a) \approx_{q-1,t(q-1)} (\mathfrak{B}, b)$. By induction, $(\mathfrak{A}, a) \equiv_{q-1}^{\text{MSO}} (\mathfrak{B}, b)$. So Duplicator wins, if she replies by b.

Suppose now that Duplicator plays a set move, say, $M \subseteq A$. Since $\mathfrak{K}$ contains at most k elements, the components of the structure $(\mathfrak{A}, M)$ belong to at most 2^k different isomorphism-types. Thus the number of q-types cannot be greater either. For each q-type θ occurring in $(\mathfrak{A}, M)$, let C_θ denote the set of components of $\mathfrak{A}$ whose q-type is θ. Duplicator chooses a set C'_θ of components of $\mathfrak{B}$ and a set of elements $M'_\theta \subseteq \bigcup_{\mathfrak{C} \in C'_\theta} C$ such that $\min\{|C_\theta|, t(q-1)\} = \min\{|C'_\theta|, t(q-1)\}$, and $\text{tp}_q(\mathfrak{C} \restriction M'_\theta \cap C) = \theta$ for each $\mathfrak{C} \in C'_\theta$. Since there are $t(q) > 2^k \cdot t(q-1)$ copies of $\mathfrak{K}$ in $\mathfrak{B}$, this is possible. Let $M' := \bigcup_\theta M'_\theta$. We have $(\mathfrak{A}, M) \approx_{q-1,t(q-1)} (\mathfrak{B}, M')$. So, by induction, $(\mathfrak{A}, M) \equiv_{q-1}^{\mathrm{MSO}} (\mathfrak{B}, M')$. Replying by M', Duplicator wins. □

Lemma 5.6.4 is an adaptation of [NdM12, Thm. 6.7] from FO to MSO. Its proof uses the previous lemma and the following analogue to Lemma 5.3.1, which can be proved like Lemma 5.3.1.

Lemma 5.6.6. *Let $q \in \mathbb{N}^+$. Let $\mathfrak{A}, \mathfrak{B} \in \mathrm{FIN}_\sigma$ be connected structures with $\mathrm{td}(\mathfrak{A}), \mathrm{td}(\mathfrak{B}) > 1$ and let $r_\mathfrak{A} \in \mathrm{roots}(\mathfrak{A})$, $r_\mathfrak{B} \in \mathrm{roots}(\mathfrak{B})$ with $\alpha(\mathfrak{A}, r_\mathfrak{A}) = \alpha(\mathfrak{B}, r_\mathfrak{B})$. Then*

$$\mathfrak{A}^{[r_\mathfrak{A}]} \equiv_q^{\mathrm{MSO}} \mathfrak{B}^{[r_\mathfrak{B}]} \implies \mathfrak{A} \equiv_q^{\mathrm{MSO}} \mathfrak{B}.$$

Proof of Lemma 5.6.4. The proof is by induction on the tree-depth d. First, we consider the claim about connected structures. If $d = 1$, then each connected structure with $\mathrm{td}(\mathfrak{A}) = 1$ has size $1 \in \mathrm{EXP}_0(q)$, i.e. we can set $\mathfrak{B} := \mathfrak{A}$. Suppose now that $d > 1$. Choose a tree-depth root $r \in \mathrm{roots}(\mathfrak{A})$. By induction, since $\mathrm{td}(\mathfrak{A}^{[r]}) \leqslant d - 1$, we obtain an induced substructure $\mathfrak{B}'$ of $\mathfrak{A}^{[r]}$ such that $|B'| \in \mathrm{EXP}_{(d-1)}(q)$ and $\mathfrak{B}' \equiv_q^{\mathrm{MSO}} \mathfrak{A}^{[r]}$. Let $\mathfrak{B}$ be the substructure of $\mathfrak{A}$ induced by $B' \cup \{r\}$, i.e. $\mathfrak{B}^{[r]} = \mathfrak{B}'$. Since $\mathfrak{A}^{[r]} \equiv_q^{\mathrm{MSO}} \mathfrak{B}^{[r]}$, we obtain that $\mathfrak{A} \equiv_q^{\mathrm{MSO}} \mathfrak{B}$ in the same way as in Lemma 5.3.1. Observe that $|B| \in \mathrm{EXP}_{(d-1)}(q)$.

Consider the case that $\mathfrak{A}$ is not connected. By the construction above, we can replace each component $\mathfrak{K}$ of $\mathfrak{A}$ by an induced substructure of $\mathfrak{K}$ on $\mathrm{EXP}_{(d-1)}(q)$ elements that has the same q-type as $\mathfrak{K}$. By the Composition Lemma, this preserves the q-type of $\mathfrak{A}$. Let $k \in \mathrm{EXP}_{(d-1)}(q)$ denote the maximum number of elements in a component of $\mathfrak{A}$ after this replacement. By Lemma 5.6.3, we know that $\mathfrak{B} \equiv_q^{\mathrm{MSO}} \mathfrak{A}$ for each induced substructure $\mathfrak{B}$ of $\mathfrak{A}$ such that $n_\tau(\mathfrak{B}) \equiv_{\wedge t} n_\tau(\mathfrak{A})$ for each q-type τ, where $t := 2^{kq}$. Since there are at most 2^k non-isomorphic components in $\mathfrak{A}$ and we have to keep at most t copies of each such component, there is such a structure $\mathfrak{B}$ with $|B| \in \mathrm{EXP}_d(q)$. □

5.6.2 From MSO to FO

With the preparations above, the proof of Theorem 5.6.1 is now very similar to the proof of Theorem 5.4.1.

Proof of Theorem 5.6.1. The proof proceeds by induction on the tree-depth d, where we also show that $|\mathcal{T}_{\sigma,q,d}| \in \mathrm{EXP}_d(q)$ and $|\mathcal{T}_{\sigma,q,d}^{\mathrm{conn}}| \in \mathrm{EXP}_{(d-1)}(q)$.

Defining types of connected structures As a first step, we prove that each q-type $\tau \in \mathcal{T}_{\sigma,q,d}^{\mathrm{conn}}$ is $\mathrm{FIN}_{\sigma,d}^{\mathrm{conn}}$-equivalent to an FO$[\sigma]$-sentence $\varphi_{\tau,d}^{\mathrm{conn}}$ such that $|\varphi_{\tau,d}^{\mathrm{conn}}| \in \mathrm{EXP}_{(d-1)}(q)$ and $\mathrm{qad}(\varphi_{\tau,d}^{\mathrm{conn}}) \leqslant 3(d-1)+1$. For $d = 1$, each structure $\mathfrak{A} \in \mathrm{FIN}_{\sigma,d}^{\mathrm{conn}}$ of type τ consists of a single element of some atomic σ-type α. The FO-sentence $\varphi_{\tau,1}^{\mathrm{conn}} := \exists x\, \alpha(x)$ then defines τ. Hence $|\varphi_{\tau,1}^{\mathrm{conn}}|$ does not depend on q, $\mathrm{qad}(\varphi_{\tau,1}^{\mathrm{conn}}) = 0$, and $|\mathcal{T}_{\sigma,q,d}^{\mathrm{conn}}| \leqslant \mathrm{EXP}_0(q)$.

Now suppose that $d > 1$ and let $\tau \in \mathcal{T}_{\sigma,q,d}^{\mathrm{conn}}$. Let $R \subseteq \mathcal{T}_{\tilde{\sigma},q,d-1} \times 2^\sigma$ be a set that contains (θ, α) iff there is a structure $\mathfrak{B} \in \mathrm{FIN}_{\tilde{\sigma},d}^{\mathrm{conn}}$ with $\mathrm{tp}_q(\mathfrak{B}) = \tau$ which contains a tree-depth root $r \in \mathrm{roots}(\mathfrak{B})$ such that $\alpha(\mathfrak{B}, r) = \alpha$ and $\mathrm{tp}_q(\mathfrak{B}^{[r]}) = \theta$. Observe that, as a consequence of

Lemma 5.6.6, for each $\mathfrak{A} \in \mathrm{FIN}^{\mathrm{conn}}_{\sigma,d}$, we have $\mathrm{tp}_q(\mathfrak{A}) = \tau$ iff $(\mathrm{tp}_q(\mathfrak{A}^{[r]}), \alpha(\mathfrak{A}, r)) \in R$ for some $r \in \mathrm{roots}(\mathfrak{A})$. Now consider a q-type $\theta \in \mathcal{T}^{\mathrm{conn}}_{\tilde{\sigma},q,d-1}$ and let $\varphi_{\theta,d-1}$ be the FO$[\tilde{\sigma}]$-sentence, given by induction, which is equivalent to θ on $\mathrm{FIN}^{\mathrm{conn}}_{\tilde{\sigma},d-1}$. As a consequence of Lemma 5.3.2, we obtain that for all structures $\mathfrak{A} \in \mathrm{FIN}^{\mathrm{conn}}_{\sigma,d}$ with $\mathrm{td}(\mathfrak{A}) > 1$ and all tree-depth roots $r \in \mathrm{roots}(\mathfrak{A})$, we have $\mathfrak{A} \models \mathcal{I}(\varphi_{\theta,d-1})(r)$ iff $\mathrm{tp}_q(\mathfrak{A}^{[r]}) = \theta$.

Altogether, we obtain that the following FO$[\sigma]$-sentence is equivalent to τ on $\mathrm{FIN}^{\mathrm{conn}}_{\sigma,d}$:

$$\varphi^{\mathrm{conn}}_{\tau,d} := (\mathrm{td} \leqslant 1 \wedge \varphi_{\tau,d-1}) \ \vee \bigvee_{(\theta,\alpha)\in R} \exists x \left(\mathrm{roots}_d(x) \ \wedge \ \alpha(x) \ \wedge \ \mathcal{I}(\varphi_{\theta,d-1})(x) \right).$$

Recall that, by induction, $|\mathcal{I}(\varphi_{\theta,d-1})| \in \mathrm{EXP}_{(d-1)}(q)$ and $|\mathcal{T}_{\tilde{\sigma},q,d-1}| \in \mathrm{EXP}_{(d-1)}(q)$. Hence, $|R| \in \mathrm{EXP}_{(d-1)}(q)$. Altogether, we obtain that $|\varphi^{\mathrm{conn}}_{\tau,d}| \in \mathrm{EXP}_{(d-1)}(q)$. Using Lemma 5.6.6, we conclude that $|\mathcal{T}^{\mathrm{conn}}_{\sigma,q,d}| \leqslant 2^{\sigma} \cdot |\mathcal{T}_{\tilde{\sigma},q,d-1}| \in \mathrm{EXP}_{(d-1)}(q)$. By induction, $\mathrm{qad}(\mathcal{I}(\varphi_{\theta,d-1})) \leqslant 3(d-1)$. Hence, $\mathrm{qad}(\varphi^{\mathrm{conn}}_{\tau,d}) \leqslant 3(d-1)+1$.

Structures with multiple components Consider an MSO$[\sigma]$-sentence φ. Let $\mathcal{T}^{\mathrm{conn}}_{\sigma,q,d} := \{\tau_1, \ldots, \tau_\ell\}$, where $\ell := |\mathcal{T}^{\mathrm{conn}}_{\sigma,q,d}|$. Let $t := t(d,q) \in \mathrm{EXP}_d(q)$ be given by Lemma 5.6.2. Let Φ be the set that contains the formulae $\varphi_i := \varphi^{\mathrm{conn}}_{d,\tau_i}$ for each $i \in [1,\ell]$. Hence, $\bar{n}_\Phi(\mathfrak{A}) \equiv_{\wedge t} \bar{n}_{\mathcal{T}^{\mathrm{conn}}_{\sigma,q,d}}(\mathfrak{A}) = \bar{n}_{\mathcal{T}_{\sigma,q}}(\mathfrak{A})$ for each $\mathfrak{A} \in \mathrm{FIN}_{\sigma,d}$. Let $R \subseteq [0,t]^\ell$ be a set such that $\bar{n} \in R$ iff there exists a model $\mathfrak{A} \in \mathrm{FIN}_{\sigma,d}$ of φ with $[\bar{n}_\Phi(\mathfrak{A})]_{\wedge t} = \bar{n}$. Using Lemma 5.6.2, we obtain that $\mathfrak{A} \models \varphi$ iff $[\bar{n}_\Phi(\mathfrak{A})]_{\wedge t} \in R$, for each $\mathfrak{A} \in \mathrm{FIN}_{\sigma,d}$. Hence, the FO$[\sigma]$-sentence $\psi := \psi^\Phi_R$ of Lemma 5.4.11 is equivalent to ψ on $\mathrm{FIN}_{\sigma,d}$.

Regarding the size of ψ, note that Lemma 5.6.2 implies that $|R| \leqslant |\mathcal{T}^{\mathrm{conn}}_{\sigma,q,d}| \leqslant [0,t]^\ell$. Since

$$\begin{aligned} t^\ell \ \in \ (\mathrm{EXP}_d(q))^{\mathrm{EXP}_{(d-1)}(q)} &= (2^{\mathrm{EXP}_{(d-1)}(q)})^{\mathrm{EXP}_{(d-1)}(q)} \\ &= 2^{\mathrm{EXP}_{(d-1)}(q) \cdot \mathrm{EXP}_{(d-1)}(q)} \\ &\subseteq 2^{\mathrm{EXP}_{(d-1)}(q)} \ = \ \mathrm{EXP}_d(q) \end{aligned}$$

we obtain, by the construction of ψ according to Lemma 5.4.11, that

$$\begin{aligned} |\psi| &\leqslant c \cdot |\Phi|^2 \cdot |R| \cdot t^2. \\ &\in \mathrm{EXP}_{(d-1)}(q)^2 \cdot \mathrm{EXP}_d(q) \cdot \mathrm{EXP}_d(q)^2 \\ &\subseteq \mathrm{EXP}_d(q), \end{aligned}$$

and $\mathrm{qad}(\psi) \leqslant \mathrm{qad}(\Phi) + 2 \leqslant 3d$. □

5.7 Lower bounds

In this section, we consider lower bounds for the dependence of the height of the exponential tower which bounds the size of the formulae constructed in Theorem 5.6.1 and Theorem 5.4.1 on the tree-depth d. For the MSO-result (Theorem 5.6.1), we can show that the linear dependence is necessary. For the $<$-inv-FO-result (Theorem 5.4.1), we establish only a very weak lower bound which shows that the height cannot be constant.

5.7.1 MSO

We prove the following theorem.

Theorem 5.7.1. *There is a signature σ and a constant $c \in \mathbb{N}^+$ such that for each $d \in \mathbb{N}^+$ there is an* $\mathrm{MSO}[\sigma]$*-sentence φ_d of size $|\varphi_d| \leqslant cd$ such that each* $\mathrm{FO}[\sigma]$*-sentence ψ_d that is* $\mathrm{FIN}_{\sigma,d}$*-equivalent to φ_d has size $|\psi_d| \leqslant \mathsf{tower}(d)$.*

Theorem 5.7.1 establishes that the linear dependence on the tree-depth achieved in Theorem 5.6.1 is necessary. This can be stated precisely as follows.

Corollary 5.7.2. *There is a signature σ for which there is no real number $0 \leqslant \epsilon < 1$ such that for each $d \in \mathbb{N}^+$ and each* $\mathrm{MSO}[\sigma]$*-sentence φ there is an* $\mathrm{FIN}_{\sigma,d}$*-equivalent* $\mathrm{FO}[\sigma]$*-sentence ψ_d of size $|\psi_d| \leqslant \mathrm{EXP}_{\lfloor \epsilon d \rfloor}(\mathrm{qr}(\varphi))$.*

Proof. Consider the $\mathrm{MSO}[\sigma]$-sentences φ_d, for each $d \in \mathbb{N}^+$, and the constant c of Theorem 5.7.1. Suppose to the contrary that there exists such a constant ϵ. Then there is a constant k such that for all $d \in \mathbb{N}^+$ there is an $\mathrm{FO}[\sigma]$-sentence ψ_d which is $\mathrm{FIN}_{\sigma,d}$-equivalent to φ_d and whose size is

$$|\psi_d| \leqslant \exp_{\lfloor \epsilon d \rfloor}(\mathrm{qr}(\varphi)^k) \leqslant \exp_{\lfloor \epsilon d \rfloor}((cd)^k).$$

For sufficiently large values of d, we clearly have $\exp_{\lfloor (1-\epsilon) d \rfloor}(0) > (cd)^k$ and hence

$$\exp_{\lfloor \epsilon d \rfloor}((cd)^k) < \exp_{\lfloor \epsilon d \rfloor}(\exp_{\lfloor (1-\epsilon) d \rfloor}(0)) \leqslant \exp_d(0).$$

This means that $|\psi_d| < \exp_d(0) = \mathsf{tower}(d)$ — a contradiction to Theorem 5.6.1. □

The proof of Theorem 5.7.1 uses an encoding of large natural numbers n by shallow trees $\mathrm{enc}(n)$ from [FG06, chapter 10.3]. Here, by *trees*, we mean (unranked) directed trees which are rooted, i.e. trees which contain a root vertex from which all edges point away. The encoding is defined inductively as follows:

- $\mathrm{enc}(0)$ is the one-node tree.
- For $n \geqslant 1$, the tree $\mathrm{enc}(n)$ is obtained by creating a new root and attaching to it all trees $\mathrm{enc}(i)$ such that the i-th bit in the binary representation of n is 1.

Note that a tree encodes a number with respect to this encoding iff there are no two distinct isomorphic subtrees whose roots are children of the same vertex. But we would like to assign a natural number to each tree. To this end, we reduce each tree $\mathfrak{T}$ in a bottom-up way to a tree $\mathrm{num}(\mathfrak{T})$ that encodes a number:

- $\mathrm{num}(\mathfrak{T}) := \mathfrak{T}$ if $\mathrm{height}(\mathfrak{T}) = 1$, i.e. $\mathfrak{T} \cong \mathrm{enc}(0)$.
- If $\mathrm{height}(\mathfrak{T}) > 1$, select one tree $\mathfrak{T}_1, \ldots, \mathfrak{T}_k$ of each isomorphism type that occurs among the immediate subtrees of the root of $\mathfrak{T}$. Define $\mathrm{num}(\mathfrak{T})$ to be a tree whose root has children whose rooted subtrees are $\mathrm{num}(\mathfrak{T}_1), \ldots, \mathrm{num}(\mathfrak{T}_k)$.

Throughout the following section, we let $\sigma := \{E, R, B\}$, where E is a binary and R, B are unary relation symbols. Here, in contrast to Chapter 4, we consider trees as directed graphs. That is, a tree is a $\{E\}$-structure $\mathfrak{T}$ where $E^{\mathfrak{T}}$ is the edge relation of the tree and a *coloured tree* is a finite σ-structure $(\mathfrak{T}, R^{\mathfrak{T}}, B^{\mathfrak{T}})$, where $\mathfrak{T}$ is a tree and $R^{\mathfrak{T}}, B^{\mathfrak{T}}$ (the *red* and the *blue* vertices of $\mathfrak{T}$) form a partition of the vertex set of the tree. Structures whose components are (coloured) trees are called (coloured) *forests*. The *height* $\mathrm{height}(\mathfrak{T})$ of a (coloured) tree $\mathfrak{T}$ is the maximum number of vertices on a path from the root of $\mathfrak{T}$ to a leave of $\mathfrak{T}$. The *height* $\mathrm{height}(\mathfrak{F})$ of a (coloured) forest $\mathfrak{F}$ is the maximum height of its components.

From the proof of [FG06, Lemma 10.21][11], we obtain the following lemma.

Lemma 5.7.3. *There is a constant $c \in \mathbb{N}^+$ such that for each $d \in \mathbb{N}^+$, there is an* FO$[E]$-*formula $\mathit{eq}_d(x, y)$ of size $|\mathit{eq}_d| \leqslant cd$ such that for all forests $\mathfrak{F}$ with $\mathrm{height}(\mathfrak{F}) \leqslant d$ and all trees $\mathfrak{T}_1, \mathfrak{T}_2$ of $\mathfrak{F}$ with roots u_1, u_2, respectively, we have:*

$$\mathfrak{F} \models \mathit{eq}_d(u_1, u_2) \iff \mathrm{num}(\mathfrak{T}_1) = \mathrm{num}(\mathfrak{T}_2).$$

Note that $\mathrm{height}(\mathrm{enc}(n)) \leqslant d$ provided that $n < \mathsf{tower}(d)$. For each $d \geqslant 1$, let $\mathfrak{F}_d$ denote a coloured forest that contains exactly the trees $\mathrm{enc}(0), \ldots, \mathrm{enc}(\mathsf{tower}(d) - 1)$ whose vertices all are coloured red, let $\mathfrak{T}_d$ denote a coloured tree with $\mathrm{height}(\mathfrak{T}_d) \leqslant d$ that contains each of the trees $\mathrm{enc}(0), \ldots, \mathrm{enc}(\mathsf{tower}(d) - 1)$ as subtrees (e.g. a full $\mathsf{tower}(d-1)$-ary tree) and where all vertices are blue, and let $\mathfrak{F}_d^n$ denote the disjoint union of $\mathfrak{F}_d$ and n disjoint copies of $\mathfrak{T}_d$, for each $n \geqslant 0$.

Lemma 5.7.4. *There is a constant $c \in \mathbb{N}^+$ such that for each $d \in \mathbb{N}^+$ there exists an* MSO$[\sigma]$-*sentence φ_d of size $|\varphi_d| \leqslant cd$ such that*

$$\mathfrak{F}_d^n \models \varphi_d \iff n \geqslant \mathsf{tower}(d),$$

for each $n \in \mathbb{N}$.

Proof. Let $d \in \mathbb{N}^+$ and let $\mathrm{eq}_d(x, y, M)$ be the relativisation of the FO$[E]$-formula of Lemma 5.7.3 to a set variable M. Let $\mathsf{conn}(M)$ be an MSO$[E]$-formula which states in a forest $\mathfrak{F}$ that for each tree $\mathfrak{T}$ of $\mathfrak{F}$, the structure $\mathfrak{T}{\restriction}M$ induced by M in $\mathfrak{T}$ is connected, i.e. a tree. Let $\mathsf{root}(x, M)$ state that x is a root in the subforest induced by M. We can assume that the size of $\mathsf{conn}(M)$ and $\mathsf{root}(x, M)$ is independent of d. Now let φ_d be the following sentence:

$$\exists M \Big(\mathsf{conn}(M) \ \wedge \ \forall x \ \big(R(x) \wedge \mathsf{root}(x)\big) \to \\ \exists y \ \big(\mathsf{root}(y, M) \ \wedge \ B(y) \ \wedge \ \mathrm{eq}_h(x, y, M)\big)\Big).$$

First we argue that $n \geqslant \mathsf{tower}(d)$ implies $\mathfrak{F}_d^n \models \varphi_d$. By definition, the red trees contained in $\mathfrak{F}_d^n$ are $\mathrm{enc}(0), \ldots, \mathrm{enc}(\mathsf{tower}(d)-1)$. Since $n \geqslant \mathsf{tower}(d)$, we can choose $\mathsf{tower}(d)$ pairwise

[11] [FG06, Lemma 10.21] makes the assumption that $\mathfrak{T}_1, \mathfrak{T}_2$ are encodings of numbers n, m to conclude that $\mathfrak{F} \models \mathrm{eq}_d(u_1, u_2) \iff n = m$, i.e. $\mathfrak{T}_1 \cong \mathfrak{T}_2$. If we drop this assumption, we obtain our variant of the lemma using exactly the same formula.

distinct copies $\mathfrak{H}_0, \dots, \mathfrak{H}_{\text{tower}(h)-1}$ of $\mathfrak{T}_d$ in $\mathfrak{F}_d^n$. Since all trees $\text{enc}(0), \dots, \text{enc}(\text{tower}(d) - 1)$ occur as subtrees of $\mathfrak{T}_d$, for each $i \in [\text{tower}(d)]$ there is a set $M_i \subseteq H_i$ such that $(\mathfrak{H}_n \upharpoonright M_n) \upharpoonright \{E\} \cong \text{enc}(i)$. The set $M := M_1 \cup \dots \cup M_n$ witnesses that $\mathfrak{F}_d^n \models \varphi_d$. Now we show that $\mathfrak{F}_d^n \models \varphi_d$ implies $n \geqslant \text{tower}(d)$. Let $M \subseteq F_d^n$ witness that $\mathfrak{F}_d^n \models \varphi_d$. The forest $\mathfrak{F}_d^n$ contains trees $\text{enc}(0), \dots, \text{enc}(\text{tower}(d) - 1)$ whose vertices are all red. Hence, and according to the choice of M and the choice of $\text{eq}_d(x, y, M)$, for each $i \in [0, \text{tower}(d) - 1]$ there is a blue copy $\mathfrak{T}$ of $\mathfrak{T}_d$ in $\mathfrak{F}_d^n$ such that $\text{num}(\mathfrak{T} \upharpoonright M) = \text{num}(\text{enc}(i)) = i$. Hence, $\mathfrak{F}_d^n$ must contain at least $\text{tower}(d)$ copies of $\mathfrak{T}_d$, because M induces at most one tree in each copy of $\mathfrak{T}_d$. □

Using Lemma 5.7.4, we can easily finish the proof of Theorem 5.7.1.

Proof of Theorem 5.7.1. A standard game argument shows that FO-sentences of quantifier-rank q cannot distinguish $\mathfrak{F}_d^q$ from $\mathfrak{F}_d^{q+1}$. The winning strategy for Duplicator in the q-round game is to answer each move of Spoiler to a copy of $\mathfrak{F}_d$ by a move to the same node in the copy of $\mathfrak{F}_d$ in the other structure, and to answer each move to a copy of $\mathfrak{T}_d$ by a move to the same element of a copy of $\mathfrak{T}_d$ in the other structure, making sure that the elements a_i and a_j selected in $\mathfrak{F}_d^q$ in rounds $i, j \leqslant q$ belong to a common copy of $\mathfrak{T}_d$ in $\mathfrak{F}_d^q$ iff the corresponding elements b_j and b_k selected in $\mathfrak{F}_d^{q+1}$ belong to a common copy of $\mathfrak{T}_d$. This is clearly possible since the number of copies of $\mathfrak{T}_d$ in both structures is at least q.

Hence, for an FO-sentence ψ_d that is equivalent to the MSO-sentence φ_d of Lemma 5.7.4, we must have $|\psi_d| \geqslant \text{qr}(\psi_d) \geqslant \text{tower}(d)$. □

5.7.2 Order-invariant FO

Throughout this section, E denotes a binary relation symbol. We prove the following theorem.

Theorem 5.7.5. *There is a constant $c \in \mathbb{N}^+$ such that for each $d \in \mathbb{N}^+$ there exists an $<$-inv-FO$[E]$-sentence φ_d of size $|\varphi| \leqslant cd^2$ such that each FO$[E]$-sentence ψ_d that is FIN-equivalent to φ_d must have size $|\psi_d| \geqslant \text{tower}(d-1)$.*

The proof of Theorem 5.7.5 rests on the following theorem which is implicit in the proof of [GS05, Theorem 6.2].

Theorem 5.7.6. *There is a constant c such that for each $d \in \mathbb{N}^+$, there is an MSO$[<]$-sentence χ_d of size $|\chi_d| \leqslant cd^2$ such that $([n], <) \models \chi_d$ iff $n \geqslant \text{tower}(d)$, for each $n \in \mathbb{N}^+$.*

For each $n \in \mathbb{N}$, we define a directed graph $\mathfrak{P}_n := (2^{[n]}, E^{\mathfrak{P}_n})$ where $E^{\mathfrak{P}_n}$ is the subset relation on $[n]$. A structure which is isomorphic to $\mathfrak{P}_n$ for some $n \in \mathbb{N}$ is called a *powerset structure*. Note that we consider the set $[n]$ as part of $\mathfrak{P}_n$ by identifying each $x \in [n]$ with the singleton set $\{x\}$. We use a standard construction to turn the MSO$[<]$-sentence of Theorem (5.7.6) which speaks about linear orders into an $<$-inv-FO$[\{E\}, <]$-sentence which speaks about the singleton sets in powerset structures.

Corollary 5.7.7. *There is a $c \in \mathbb{N}$ such that for each $d \in \mathbb{N}^+$, there is an $<$-inv-FO$[E, <]$-sentence φ_d of size $|\chi_d| \leqslant cd^2$ such that $\mathfrak{P}_n \models \varphi_d$ iff $n \geqslant \text{tower}(d)$, for each $n \in \mathbb{N}^+$.*

We present only a rough sketch for the proof of the corollary, since the method is exactly the same as in Gurevich's example which shows that $<$-inv-FO is more expressive than FO on finite structures. This example is presented in full detail in [Lib04].

Proof sketch. For each $d \in \mathbb{N}$, we construct φ_d from the sentence χ_d of Theorem 5.7.6 using the following standard construction. There is an FO[E]-formula $\varphi_{\text{sing}}(x)$ which is true for an element Y of a powerset structure $\mathfrak{P}_n$ iff Y is a singleton set. We modify χ_d as follows. We replace each first-order quantifier $\exists x\, \xi$ by $\exists x\, \varphi_{\text{sing}}(x) \wedge \xi$ and each atomic formula of the form $y \in X$ by $E(y, X)$. We let φ_d be the conjunction of this modified sentence and a sentence which defines the class of all powerset structures on the class of all finite structures. Such a sentence can be obtained from order-theoretic axioms for Boolean algebras, since it is well-known that every finite Boolean algebra is isomorphic to a Boolean algebra of subsets of a finite set. $\square$

The proof of Gurevich's example in the book [Lib04] contains the following result.

Lemma 5.7.8 ([Lib04, Claim 5.7]). *$\mathfrak{P}_{2^q} \equiv_q^{\text{FO}} \mathfrak{P}_{2^q+\ell}$, for all $q, \ell \in \mathbb{N}$.*

Now we can finish the proof of Theorem 5.7.5.

Proof of Theorem 5.7.5. For each $d \in \mathbb{N}$, we consider the sentence φ_d of Corollary 5.7.7. By Lemma 5.7.8, we have

$$\mathfrak{P}_{2^{\text{tower}(d-1)-1}} \equiv^{\text{FO}}_{\text{tower}(d-1)-1} \mathfrak{P}_{2^{\text{tower}(d-1)}} = \mathfrak{P}_{\text{tower}(d)}.$$

Assume that ψ_d is equivalent to φ_d. That is, $\mathfrak{P}_{2^{\text{tower}(d-1)-1}} \not\models \psi_d$ and $\mathfrak{P}_{\text{tower}(d)} \models \psi_d$. Hence, we must have $|\psi_d| \geqslant \text{qr}(\psi_d) > \text{tower}(d-1) - 1$. $\square$

A slight strengthening of Theorem 5.7.5 shows that the height of the exponential tower in Theorem 5.4.1 cannot be a constant which is independent of the tree-depth.

Corollary 5.7.9. *There is no constant D such that for each $<$-inv-FO[E]-sentence φ and each $d \in \mathbb{N}$, there is an $\text{FIN}_{\{E\},d}$-equivalent FO[E]-sentence ψ_d whose size is $|\psi_d| \in \text{EXP}_D(\text{qr}(\varphi))$.*

Proof. Suppose to the contrary that there is such a constant D and consider the family of $<$-inv-FO[E]-sentences φ_d of Corollary 5.7.7. Then there is an constant k such that for all $d, d' \in \mathbb{N}^+$, there is an $\text{FIN}_{\{E\},d'}$-equivalent sentence $\psi_{d,d'}$ for φ_d of size

$$|\psi_{d,d'}| \leqslant \exp_D(\text{qr}(\varphi_d)^k) \leqslant \exp_D((cd^2)^k),$$

for some constant c which is independent of d. It is clearly possible to choose d such that $\exp_D((cd^2)^k) < \text{tower}(d-1)$. Let $d' := \text{tower}(d+1)$ and $\psi := \psi_{d,d'}$. The tree-depth of a structure is at most its cardinality. Hence, $\text{td}(\mathfrak{P}_n) \leqslant 2^n$. If $n \leqslant \text{tower}(d)$, we obtain $\text{td}(\mathfrak{P}_n) \leqslant d'$ and thus $\mathfrak{P}_n \in \text{FIN}_{\{E\},d'}$. Since ψ is $\text{FIN}_{\{E\},d'}$-equivalent to φ_d, we have $\mathfrak{P}_{2^{\text{tower}(d-1)-1}} \not\models \psi$ and $\mathfrak{P}_{\text{tower}(d)} \models \psi$. Since $|\psi| < \text{tower}(d-1)$, this contradicts Lemma 5.7.8. $\square$

5.8 Conclusion

In the preceding chapter, we have studied expressivity and succinctness questions for FO and MSO and the order-invariant formulae of these logics on structures of bounded tree-depth.

We have established that, on structures of tree-depth d, each $<$-inv-FO- and each MSO-sentence can be translated to an at most d-fold-exponentially larger equivalent FO-sentence. For MSO, we have proved a lower bound which shows that the dependence on d in this result is necessary for rooted and coloured trees. This also shows that MSO is non-elementarily more succinct than FO on the class of all rooted and coloured trees. From our result, we have derived that $<$-inv-FO is non-elementarily more succinct than FO on finite structures. This shows that the height of the exponential tower in the translation from $<$-inv-FO to FO on bounded tree-depth structures cannot be constant. Unfortunately, we were not able to show that a linear dependence as achieved by our upper bound is necessary.

We have shown that each $<$-inv-MSO-sentence is equivalent to an FO+MOD-sentence on bounded tree-depth structures. We have not considered the corresponding succinctness question since it follows from the results of [GS05] that $<$-inv-MSO is non-elementary more succinct than FO+MOD on structures over the empty signature. For this, it is necessary that the definition of the length of FO+MOD-formulae takes the length of the binary representation of the numbers which occur in modulo-counting quantifiers into account. If we chose not do this, i.e. we would essentially define the size of a formula as the number of nodes in it syntax tree, the author believes that it should be possible to obtain elementary bounds on the size of the constructed FO+MOD-formulae, similarly as in the results for MSO and $<$-inv-FO. However, the chosen representation seems more natural. The author also believes that it is possible to obtain elementary upper bounds for the translation from CMSO to FO+MOD, using our definition of formula length.

It follows from results of [SS10] that $+$-inv-FO has the same expressive power as $\mathrm{FO}_{\mathrm{card}}$ on structures of tree-depth 1. It would be interesting to know if this result can be extended to structures of arbitrary bounded tree-depth. The apparent difficulty is that the composition techniques which we used to obtain the results of this chapter are not available in the presence of addition.

One motivation to study bounded tree-depth graphs is the role of these graphs in the theory of sparse graphs [NdM12]. This link has been exploited in several results about the algorithmic behaviour of logics on sparse structures. Can the approach that we used to obtain our results on $<$-inv-FO-sentences on bounded tree-depth structures be used to obtain results about such sentences on more general classes of sparse structures?

Chapter 6

Final remarks

This thesis has obtained results concerning order-invariant, addition-invariant, and $\mathcal{ARB}$-invariant formulae of FO and its extensions. Several questions for further research which arise from this work have been noted throughout the thesis and in the "Conclusion"-sections at the end of chapters 3, 4, and 5. We recollect some of these questions which we consider particularly interesting. We also include several open questions which are not immediately connected to the results of this thesis.

Question 1: Are all $+$-inv-FO-definable tree languages regular?

A positive answer would mean that our characterisation of regular $+$-inv-FO-definable tree languages extends to all $+$-inv-FO-definable tree languages. Note that this question is also open for the restricted case of word languages [SS10].

Question 2: Are $+$-inv-FO-definable queries constantly Gaifman-local?

This questions was raised in [GS00]. From this paper, we know that order-invariantly definable queries are Gaifman-local with constant locality radius. The paper [AvMSS12] established that all $\mathcal{ARB}$-inv-FO-definable queries are Gaifman-local with a polylogarithmic locality radius and that a polylogarithmic radius is necessary for $\{+, *\}$-inv-FO-formulae.

Question 3: Is $<$-inv-FO Hanf-local with a constant locality radius?

Hanf locality with a constant locality radius on trees follows from [Nie07], [BS09b], but apart from this, nothing seems to be known.

Question 4: Is there an $<$-inv-FO-definable class which is not MSO-definable?

This question was posed by Benedikt and Segoufin [BS09b] (for CMSO instead of MSO). Recently, Elberfeld, Frickenschmidt and Grohe obtained strong results showing that, for graph classes with excluded minors, every $<$-inv-FO-definable subclass is also MSO-definable and every $<$-inv-MSO-definable class is CMSO-definable. Ganzow and Rubin [GR08] proved that there is an $<$-inv-MSO-definable class which is not CMSO-definable. The usual approach to lift $<$-inv-MSO-sentences to $<$-inv-FO-sentences on powerset structures does not seem to help to answer this question, since MSO becomes too powerful on powerset structures.

Question 5: Is there a decidable characterisation of the $<$-inv-FO+MOD$_2$-definable languages?

In Section 3.6, we have shown that there are $<$-inv-FO+MOD$_2$-definable word languages which are not FO+MOD$_p$-definable, for any modulus p. It would be interesting to gain a better understanding of the expressive power of $<$-inv-FO+MOD$_2$ on words.

Question 6: Is the d-fold exponential dependence of the FO-formula size on the size of the $<$-inv-FO-formula in Theorem 5.4.1 necessary?

Question 7: Is $<$-inv-FO^2 contained in FO?

Together with Thomas Zeume, the author has considered order-invariant sentences of the two-variable fragment of first-order logic ($<$-inv-FO^2). The paper [ZH16] shows that order-invariance of FO^2-sentences is decidable. This work has led to the above question. It is very easy to see that $<$-inv-FO^2 is more expressive than FO^2 (cf. [ZH16]), but all examples known to the author which show that $<$-inv-FO is more expressive than FO seem to require more than two variables.

Bibliography

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge Univ. Press, 2009.

[Ajt83] Miklos Ajtai. Σ_1^1-formulae on finite structures. *Annals of Pure and Applied Logic*, 24(1):1–48, 1983.

[AU79] Alfred V. Aho and Jeffrey D. Ullman. Universality of data retrieval languages. In *Proceedings of the 6th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages*, POPL '79, pages 110–119. ACM, 1979.

[AvMSS12] Matthew Anderson, Dieter van Melkebeek, Nicole Schweikardt, and Luc Segoufin. Locality from circuit lower bounds. *SIAM Journal on Computing*, 41(6):1481–1523, 2012.

[BDK12] Adam Bouland, Anuj Dawar, and Eryk Kopczyński. On tractable parameterizations of graph isomorphism. In *Parameterized and Exact Computation: 7th International Symposium, IPEC 2012, Proceedings*, pages 218–230. Springer, 2012.

[BIS90] David A. Mix Barrington, Neil Immerman, and Howard Straubing. On uniformity within NC^1. *Journal of Computer and System Sciences*, 41(3):274–306, 1990.

[BK12] Benedikt Bollig and Dietrich Kuske. An optimal construction of Hanf sentences. *Journal of Applied Logic*, 10(2):179–186, 2012.

[BL16] Pablo Barceló and Leonid Libkin. Order-invariant types and their applications. *Logical Methods in Computer Science*, 12(1), 2016.

[BP89] Danièle Beauquier and Jean-Éric. Pin. Factors of words. In *Proceedings of ICALP'89*, volume 372 of *Lecture Notes in Computer Science*, pages 63–79. Springer, 1989.

[BP16] Mikołaj Bojańczyk and Michał Pilipczuk. Definability equals recognizability for graphs of bounded treewidth. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16*. ACM, 2016.

[BS05] Michael Benedikt and Luc Segoufin. Regular tree languages definable in FO. In *STACS 2005, 22nd Annual Symposium on Theoretical Aspects of Computer Science, Proceedings*. Springer, 2005.

[BS09a] Michael Benedikt and Luc Segoufin. Regular tree languages definable in FO and in FO_{mod}. *ACM Transactions on Computational Logic*, 11(1), 2009.

[BS09b] Michael Benedikt and Luc Segoufin. Towards a characterization of order-invariant queries over tame graphs. *Journal of Symbolic Logic*, 74(1):pp. 168–186, 2009.

[BSS12] Mikołaj Bojánczyk, Luc Segoufin, and Howard Straubing. Piecewise testable tree languages. *Logical Methods in Computer Science*, 8(3), 2012.

[Büc60] J. Richard Büchi. Weak second-order arithmetic and finite automata. *Mathematical Logic Quarterly*, 6:66–92, 1960.

[CDG$^+$07] Hubert Comon, Max Dauchet, Rémi Gilleron, Florent Jacquemard, Denis Lugiez, Christof Löding, Sophie Tison, and Marc Tommasi. Tree automata techniques and applications. Available at `http://tata.gforge.inria.fr/`, 2007. Release: October, 12th 2007.

[CE12] Bruno Courcelle and Joost Engelfriet. *Graph Structure and Monadic Second-Order Logic – A Language-Theoretic Approach*. Cambridge University Press, 2012.

[CH82] Ashok Chandra and David Harel. Structure and complexity of relational queries. *Journal of Computer and System Sciences*, 25(1):pp. 99–128, 1982.

[Cod70] Edgar Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, June 1970.

[Cod72] Edgar Codd. Relational completeness of data base sublanguages. In *Database Systems*, pages 65–98. Prentice-Hall, 1972.

[Cou91] Bruno Courcelle. The monadic second-order logic of graphs v: on closing the gap between definability and recognizability. *Theoretical Computer Science*, 80:pp. 153–202, 1991.

[Cou96] Bruno Courcelle. The monadic second-order logic of graphs x: linear orderings. *Theoretical Computer Science*, 160(1–2):pp. 87–143, 1996.

[Daw10] Anuj Dawar. Homomorphism preservation on quasi-wide classes. *Journal of Computer and System Sciences*, 76(5):324–332, 2010.

[DLM07] Arnaud Durand, Clemens Lautemann, and Malika More. Counting results in weak formalisms. In *Circuits, Logic, and Games*, number 06451 in Dagstuhl Seminar Proceedings. Schloss Dagstuhl, Germany, 2007.

[Don70] John Doner. Tree acceptors and some of their applications. *Journal of Computer and System Sciences*, 4(5):406–451, 1970.

[EEH14] Kord Eickmeyer, Michael Elberfeld, and Frederik Harwath. Expressivity and succinctness of order-invariant logics on depth-bounded structures. In *Proceedings of the 39th International Symposium on Mathematical Foundations of Computer Science (MFCS 2014), Part I*, pages 256–266. Springer, 2014.

[EEH16] Kord Eickmeyer, Michael Elberfeld, and Frederik Harwath. Succinctness of order-invariant logics on depth-bounded structures. arXiv:1603.09055v1 [cs.LO], 2016.

[EF99] Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite Model Theory*. Springer, 1999.

[EFG16] Michael Elberfeld, Marlin Frickenschmidt, and Martin Grohe. Order invariance on decomposable structures. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16*. ACM, 2016.

[EFT07] Heinz-Dieter Ebbinghaus, Jörg Flum, and Wolfgang Thomas. *Einführung in die Mathematische Logik*. Spektrum Akademischer Verlag, 5. edition, 2007.

[EGT12] Michael Elberfeld, Martin Grohe, and Till Tantau. Where first-order and monadic second-order logic coincide. In *Proceedings of the 27th Annual IEEE Symposium on Logic in Computer Science, LICS 2012*, pages 265–274, 2012.

[EGT16] Michael Elberfeld, Martin Grohe, and Till Tantau. Where first-order and monadic second-order logic coincide. *ACM Transactions on Computational Logic*, 17(4):25:1–25:18, September 2016.

[EJT12] Michael Elberfeld, Andreas Jakoby, and Till Tantau. Algorithmic meta theorems for circuit classes of constant and logarithmic depth. In *29th International Symposium on Theoretical Aspects of Computer Science, STACS 2012*, pages 66–77, 2012.

[Fag74] Ronald Fagin. Generalized first-order spectra, and polynomial. time recognizable sets. *SIAM-AMS Proceedings*, 7:43–73, 1974.

[FG06] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Springer, 2006.

[FSV95] Ronald Fagin, Larry J. Stockmeyer, and Moshe Y. Vardi. On Monadic NP vs. Monadic co-NP. *Information and Computation*, 120(1):78–92, 1995.

[Gai82] Haim Gaifman. On local and nonlocal properties. In *Logic Colloquium '81*, page 105–135. North-Holland, 1982.

[GH15] Jakub Gajarský and Petr Hlinený. Kernelizing MSO properties of trees of fixed height, and some consequences. *Logical Methods in Computer Science*, 11(1), 2015.

[GKS13] Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Characterisations of nowhere dense graphs (invited talk). In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2013*, pages 21–40, 2013.

[GKS14] Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding first-order properties of nowhere dense graphs. In *Symposium on Theory of Computing, STOC 2014*, pages 89–98. ACM, 2014.

[GR08] Tobias Ganzow and Sasha Rubin. Order-invariant MSO is stronger than Counting MSO in the finite. In *STACS 2008, 25th Annual Symposium on Theoretical Aspects of Computer Science, Proceedings*, LIPIcs, pages 313–324. Schloss Dagstuhl, 2008.

[Gra93] Etienne Grandjean. Linear time algorithms and np-complete problems. In *Computer Science Logic: 6th Workshop, CSL '92*, pages 248–273. Springer, 1993.

[Gro08] Martin Grohe. The quest for a logic capturing ptime. In *23rd Annual IEEE Symposium on Logic in Computer Science, 2008. LICS '08.*, pages 267–271, 2008.

[GS00] Martin Grohe and Thomas Schwentick. Locality of order-invariant first-order formulas. *ACM Transactions on Computational Logic*, 1(1):112–130, 2000.

[GS05] Martin Grohe and Nicole Schweikardt. The succinctness of first-order logic on linear orders. *Logical Methods in Computer Science*, 1(1:6):pp. 1–25, 2005.

[Gur88] Yuri Gurevich. Logic and the challenge of computer science. In Egon Börger, editor, *Current Trends in Theoretical Computer Science*, pages 1–57. Computer Science Press, 1988.

[Han65] William Hanf. Model-theoretic methods in the study of elementary logic. In Alfead Tarski John W. Addison, Leon Henkin, editor, *The Theory of Models*, page 132–145. North Holland, 1965.

[Hås87] Johan Håstad. *Computational Limitations of Small-depth Circuits*. MIT Press, Cambridge, MA, USA, 1987.

[HHI$^+$07] Joseph Y. Halpern, Robert Harper, Neil Immerman, Phokion G. Kolaitis, Moshe Y. Vardi, and Victor Vianu. On the unusual effectiveness of logic in computer science. *The Bulletin of Symbolic Logic*, 7(2):213–236, Juni 2007.

[HHS14] Frederik Harwath, Lucas Heimberg, and Nicole Schweikardt. Preservation and decomposition theorems for bounded degree structures. In *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14*, pages 49:1–49:10, 2014.

[HHS15] Frederik Harwath, Lucas Heimberg, and Nicole Schweikardt. Preservation and decomposition theorems for bounded degree structures. *Logical Methods in Computer Science*, 11(4), 2015.

[HKS16] Lucas Heimberg, Dietrich Kuske, and Nicole Schweikardt. Hanf normal form for first-order logic with unary counting quantifiers. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16*, 2016.

[HLN99] Lauri Hella, Leonid Libkin, and Juha Nurmonen. Notions of locality and their logical characterizations over finite models. *Journal of Symbolic Logic*, 64(4):1751–1773, 1999.

[How76] John M. Howie. *An introduction to semigroup theory*. Academic Press, 1976.

[HS12] Frederik Harwath and Nicole Schweikardt. Regular tree languages, cardinality predicates, and addition-invariant FO. In *29th International Symposium on Theoretical Aspects of Computer Science, STACS 2012*, LIPIcs, pages 489–500. Schloss Dagstuhl, 2012.

[HS13] Frederik Harwath and Nicole Schweikardt. On the locality of arb-invariant first-order logic with modulo counting quantifiers. In *Computer Science Logic 2013, CSL 2013*, volume 23 of *LIPIcs*, pages 363–379. Schloss Dagstuhl, 2013.

[HS16] Frederik Harwath and Nicole Schweikardt. On the locality of arb-invariant first-order formulas with modulo counting quantifiers. *Logical Methods in Computer Science*, 12(4), December 2016.

[HU79] John E. Hopcroft and Jeffrey D Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.

[Imm86] Neil Immerman. Relational queries computable in polynomial time. *Information and Control*, 68(1):86 – 104, 1986.

[Imm87] Neil Immerman. Languages that capture complexity classes. *SIAM Journal on Computing*, 16(4):760–778, 1987.

[Imm99] Neil Immerman. *Descriptive Complexity*. Springer, 1999.

[JJ98] Gareth A. Jones and Josephine M. Jones. *Elementary Number Theory*. Springer, 1998.

[Lam14] Michael Lampis. Model checking lower bounds for simple graphs. *Logical Methods in Computer Science*, 10(1), 2014.

[Lib04] Leonid Libkin. *Elements of Finite Model Theory*. Springer, 2004.

[Lot84] M. Lothaire. *Combinatorics on words*. Cambridge University Press, 1984.

[LSW01] Carsten Lutz, Ulrike Sattler, and Frank Wolter. Modal logic and the two-variable fragment. In *Computer Science Logic: 15th International Workshop, CSL 2001 10th Annual Conference of the EACSL*, pages 247–261. Springer, 2001.

[LW02] Leonid Libkin and Limsoon Wong. Lower bounds for invariant queries in logics with counting. *Theoretical Computer Science*, 288(1):153 – 180, 2002.

[Mak04] Johann A. Makowsky. Algorithmic uses of the Feferman–Vaught theorem. *Annals of Pure and Applied Logic*, 126(1–3):pp. 159–213, 2004.

[Min67] Marvin L. Minsky. *Computation: Finite and Infinite Machines*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1967.

[MP71] Robert McNaughton and Seymour A. Papert. *Counter-Free Automata*. The MIT Press, 1971.

[NdM06] Jaroslav Nešetřil and Patrice Ossona de Mendez. Tree-depth, subgraph coloring and homomorphism bounds. *European Journal of Combinatorics*, 27(6):1022–1041, August 2006.

[NdM10] Jaroslav Nešetřil and Patrice Ossona de Mendez. First order properties on nowhere dense structures. *The Journal of Symbolic Logic*, 75(3):868–887, 2010.

[NdM12] Jaroslav Nešetřil and Patrice Ossona de Mendez. *Sparsity: Graphs, Structures, and Algorithms*. Springer, 2012.

[NdM15] Jaroslav Nešetřil and Patrice Ossona de Mendez. On low tree-depth decompositions. *Graphs and Combinatorics*, 31(6):1941–1963, 2015.

[Nie05] Hannu Niemistö. On locality and uniform reduction. In *20th IEEE Symposium on Logic in Computer Science (LICS 2005), Proceedings*, pages 41–50, 2005.

[Nie07] Hannu Niemistö. *Locality and Order-Invariant Logics*. PhD thesis, Department of Mathematics and Statistics, University of Helsinki, 2007.

[Nur96] Juha Nurmonen. Counting modulo quantifiers on finite structures. *Information and Computation*, 160:484–493, 1996.

[Ott00] Martin Otto. Epsilon-logic is more expressive than first-order logic over finite structures. *Journal of Symbolic Logic*, 65(4):1749–1757, 2000.

[Pot94] Andreas Potthoff. Modulo-counting quantifiers over finite trees. *Theor. Comput. Sci.*, 126:97–112, April 1994.

[PS11] Thomas Place and Luc Segoufin. A decidable characterization of locally testable tree languages. *Logical Methods in Computer Science*, 7(4), 2011.

[Ros08] Benjamin Rossman. Homomorphism preservation theorems. *Journal of the ACM*, 55(3), 2008.

[Rot94] Joseph J. Rotman. *An Introduction to the Theory of Groups*. Springer, 4th edition, 1994.

[Sch65] Marcel-Paul Schützenberger. On finite monoids having only trivial subgroups. *Information and Control*, 8(2):190–194, 1965.

[Sch06] Nicole Schweikardt. On the expressive power of monadic least fixed point logic. *Theoretical Computer Science*, 350(2–3):325 – 344, 2006.

[Sch07] Nicole Schweikardt. An Ehrenfeucht-Fraïssé game approach to collapse results in database theory. *Information and Computation*, 205(3):311–379, 2007.

[Sch13] Nicole Schweikardt. A short tutorial on order-invariant first-order logic. In *Computer Science - Theory and Applications - 8th International Computer Science Symposium in Russia, CSR 2013, Ekaterinburg, Russia, June 25-29, 2013. Proceedings*, pages 112–126, 2013.

[Smo87] Roman Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, pages 77–82, New York, NY, USA, 1987. ACM.

[SS10] Luc Segoufin and Nicole Schweikardt. Addition-invariant FO and regularity. In *Proceedings of 25th IEEE Symposium on Logic in Computer Science (LICS'10)*, pages 285–294. IEEE, 2010.

[Str94] Howard Straubing. *Finite automata, formal logic, and circuit complexity*. Birkäuser, 1994.

[Tho96] Wolfgang Thomas. Languages, automata, and logic. In *Handbook of Formal Languages*, pages 389–455. Springer, 1996.

[TW68] James W. Thatcher and Jesse B. Wright. Generalized finite automata theory with an application to a decision problem of second-order logic. *Theory of Computing Systems*, 2:57–81, 1968.

[Var82] Moshe Y. Vardi. The complexity of relational query languages (extended abstract). In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, STOC '82, pages 137–146. ACM, 1982.

[Wil11] Ryan Williams. Guest column: A casual tour around a circuit complexity bound. *SIGACT News*, 42(3):54–76, 2011.

[ZH16] Thomas Zeume and Frederik Harwath. Order-invariance of two-variable logic is decidable. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16*, pages 807–816. ACM, 2016.

List of notation

Index

Erklärung

Hiermit erkläre ich Folgendes:

- Die Promotionsordnung der Mathematisch-Naturwissenschaftlichen Fakultät der Humboldt-Universität zu Berlin vom 18.11.2014, veröffentlicht im Amtlichen Mitteilungsblatt Nr. 126/2014, ist mir bekannt.
- Die vorliegende Dissertation wurde von mir selbstständig und nur unter Zuhilfename der von mir gemäß §7 Abs. 3 der oben genannten Promotionsordnung angegebenen Hilfsmittel verfasst.
- Ich besitze keinen Doktorgrad und habe mich auch nicht anderwärts um einen Doktorgrad beworben.

Frederik Harwath

Gelnhausen, den 29. Mai 2017